AD-A051 468 AIR FORCE ARMAMENT LAB EGLIN AFB FLA F/G 9/2
STABILITY ANALYSIS PROGRAM FOR THE HEWLETT-PACKARD 9830A CALCUL--ETC(U)
JUL 75 K A GALE, R D GREEN, J M GONZALEZ
UNCLASSIFIED AFATL-TR-75-94 NL

1 OF 2

AD
A051 468

code 23

AD-A051468

AFATL-TR-75-94

STABILITY ANALYSIS PROGRAM FOR THE HEWLETT-PACKARD 9830A CALCULATOR

SYSTEMS ANALYSIS AND SIMULATION BRANCH
GUIDED WEAPONS DIVISION

JULY 1975

FINAL REPORT: SEPTEMBER 1974 - DECEMBER 1974

Cleared for Open Publication
ADTC/OI, #1, 30 Sep 77.

AIR FORCE ARMAMENT LABORATORY

AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE

EGLIN AIR FORCE BASE, FLORIDA

DEPARTMENT OF THE AIR FORCE
AIR FORCE ARMAMENT LABORATORY (AFSC)
EGLIN AIR FORCE BASE, FLORIDA 32542

REPLY TO
ATTN OF: DLOD

9 Aug 77

SUBJECT: Distribution Statement on Technical Report

TO: DLMA

1. Regulations require review at least at two-year intervals of all technical reports with a limited distribution statement with the objective of removing the limitation as soon as conditions permit.

2. It is requested that AFATL-TR-75-94 , subject, "Stability Analysis Program for the Hewlett-Packard 9830A Calculator"

be reviewed to determine if it may be approved for "Statement A" (approved for public release, distribution unlimited) or if it should properly contain "Statement B" as delineated in par. IV,A of Attachment 1, AFR 80-45, 26 March 1971.

ALBERT T. JOSEPH, Captain, USAF
Chief, Data Services Branch

Bill —

I don't see why this was limited in the first place. Unlimited distr is OK with me.

— Steve

TO: DLODL
Statement A is appropriate for this TR
William P. Cast - DLMA OSA, 10 Aug 1977

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFATL-TR-75-94
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): STABILITY ANALYSIS PROGRAM FOR THE HEWLETT-PACKARD 9830A CALCULATOR
5. TYPE OF REPORT & PERIOD COVERED: Final Report, September - December 1974
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Kenneth A. Gale, Major, USAF; Robert D. Green, Major, USAF; Jesse M. Gonzalez
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Guided Weapons Division (DLMA), Air Force Armament Laboratory, Eglin Air Force Base, Florida 32542
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 20680001
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Armament Laboratory, Armament Development and Test Center, Eglin Air Force Base, Florida 32542
12. REPORT DATE: July 1975
13. NUMBER OF PAGES: 107
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

ACCESSION for		
NTIS	White Section	X
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A	23 Per E.G.	

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES: Available in DDC
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Hewlett-Packard 9830A Calculator; Stability Analysis
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The stability analysis pac for the Hewlett-Packard 9830 calculator is a collection of programs to aid the engineer in performing the linear analysis of control systems. One of the programs will reduce a complex block diagram of a control system to a simple equivalent forward and feedback transfer function. The second series of programs will generate the root locus, the Bode stability and frequency responses, and the time response to a step input. In all instances, the programs will generate plots in addition to numerical printouts. The programs will handle up to 10th order systems. This report contains a complete description of the programs, including flow charts and program listings, and examples of the use of the programs.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DISCLAIMER NOTICE

DEPARTMENT OF THE AIR FORCE
AIR FORCE ARMAMENT LABORATORY (AFSC)
EGLIN AIR FORCE BASE, FLORIDA 32542

REPLY TO
ATTN OF: DLOD

4 Oct 77

SUBJECT: Distribution Statement on Technical Report

TO: AF Liaison Representative
Defense Documentation Center (TSR)
Cameron Station
Alexandria, VA 22314

I certify that the following TR, which was formerly limited distribution, has been reviewed and approved for public release by the controlling office and the Information Office in accordance with AFR 80-45 and AFSC Supplement I thereto. It may be made available or sold to the general public and foreign nationals:

AFATL-TR-75-94: "Stability Analysis Program for the Hewlett-Packard 9830A Calculator"

ALBERT T. JOSEPH, Captain, USAF
Chief, Data/Services Branch

Cy: AFSC/DLXL

AU (AUL/LSE-70-230)
Maxwell AFB, AL 36112

PREFACE

This technical report is based on work performed in support of Project 20680001 at the Air Force Armament Laboratory, Armament Development and Test Center, Eglin Air Force Base, Florida 32542. The work was performed from September to December 1974 by the Systems Analysis and Simulation Branch, Guided Weapons Division.

This technical report has been reviewed and is appproved for publication.

FOR THE COMMANDER

FENDRICK J. SMITH, JR., Colonel, USAF
Chief, Guided Weapons Division

(The reverse of this page is blank)

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

SECTION I

INTRODUCTION

The objective of this effort is to develop a package of programs for the Hewlett-Packard (HP) 9830 calculator which will aid the engineer in the process of analyzing the stability and control of complex systems -- in particular, guided weapon systems. The programs presented here will aid the engineer in performing the linear analysis associated with continuous systems, and they will provide a means for debugging and checking more detailed, non-linear simulations of the system.

Performing linear analysis on a system is typically a two-step process. The first step involves the reduction of a complex block diagram, which represents the system, into a simple, equivalent block diagram. The loop solver program (LSP) performs this block diagram reduction. This program will accept a block diagram which contains up to seven individual forward transfer functions and up to four feedback transfer functions and will reduce these to a single, equivalent forward transfer function and the associated feedback transfer function.

The second step of the linear analysis involves using these equivalent transfer functions to generate a root locus plot, a Bode stability plot, a Bode response plot, a time response plot, or a variety of other plots which will yield information about the gain and phase margins, the loop frequencies and the damping, and the loop response of the system. The stability analysis program (SAP) will generate the root locus plot, the Bode stability and response plots, and the time response plot of the system.

There are two versions of the stability analysis executive program. One version is designed to take the equivalent transfer functions directly from the LSP by means of a data tape. This version is referred to as Stability Analysis Program-Tape Input (SAP-TI). The second is designed to allow the operator to enter a single forward transfer function and a single feedback transfer function by means of the keyboard, without having to resort to the LSP. This second version is referred to as the Stability Analysis Program-Keyboard Input (SAP-KI).

Due to the size of the second version of the SAP, the entire program cannot reside in memory at one time. Therefore, each SAP consists of an executive routine, which resides in memory, and three subroutines, only one of which resides in memory at any one time. The basic interaction of these programs is shown in Figure 1.

This report describes the loop solver program and the two versions of the stability analysis program, along with the three subroutines and their use. A discussion of block diagram reduction technique and stability analysis techniques is contained in Reference 1, and the HP 9830 BASIC language is presented in Reference 2.

References:

1. D'Azzo and Houpis, Feedback Control System Analysis and Synthesis. McGraw-Hill, N. Y., 1966.
2. Hewlett-Packard 9830A Calculator Operating and Programming Manual. Hewlett-Packard Company, Loveland, Colorado, 1973.

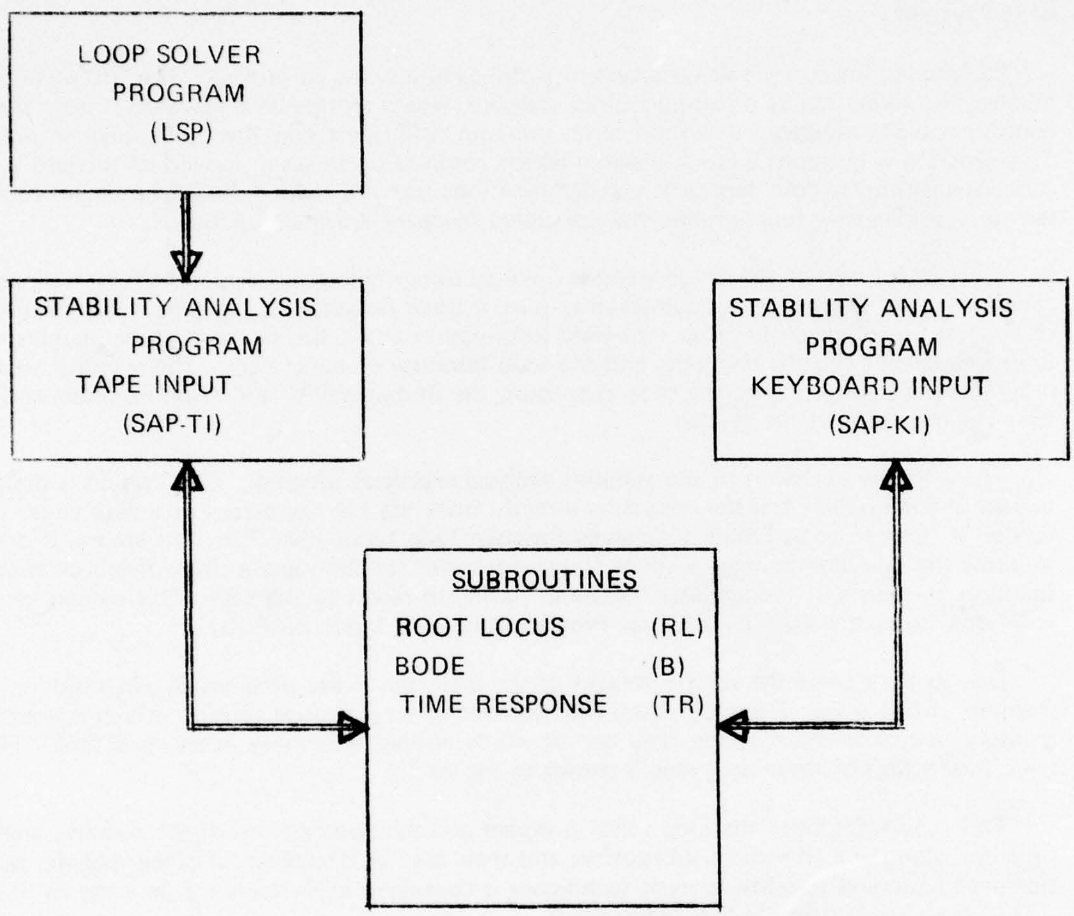

Figure 1. Program Interaction Chart

SECTION II

PROGRAM REQUIREMENTS AND RESTRICTIONS

1. REQUIREMENTS

The programs described in this report require a HP 9830 calculator with the following options:

HP 9862A Calculator Plotter

Plotter ROM

String Variable ROM

Matrix ROM

8K words of memory

2. RESTRICTIONS

The restrictions enumerated below apply, in general, to all of the programs and their operation. In those instances where the restriction applies to a single program, it is so noted.

a. The system block diagram must be reduced to the form shown in Figure 2 prior to being entered into the loop solver program.

b. The data storage file length must be at least 2000 words.

c. The order of the total system being processed must not exceed 10th. (Note: The order is determined by adding the order of the denominator of each forward and feedback transfer function.)

d. The programs assume negative feedback. If the feedback is positive, a negative sign must be entered as a part of the feedback transfer function.

e. All polynomials are entered from the lowest order coefficient to the highest order coefficient. Constants are entered as polynomials of order zero.

f. In the loop solver program, all transfer functions are initialized to zero. However, if it is necessary to zero out a non-zero transfer function, enter a zero in the numerator and a one in the denominator. The program does not recognize 0/0! This restriction also applies to entering a feedback transfer function of zero in the keyboard input version of the SAP-KI.

g. When the calculator asks a question which requires a letter response, e.g., "Y" or "N" for "yes" or "no", the appropriate letter must be entered. Any other input, including the command "STOP", will result in an error, i.e., ERROR 77. This error is a recoverable error, and execution will continue when the appropriate letter is entered.

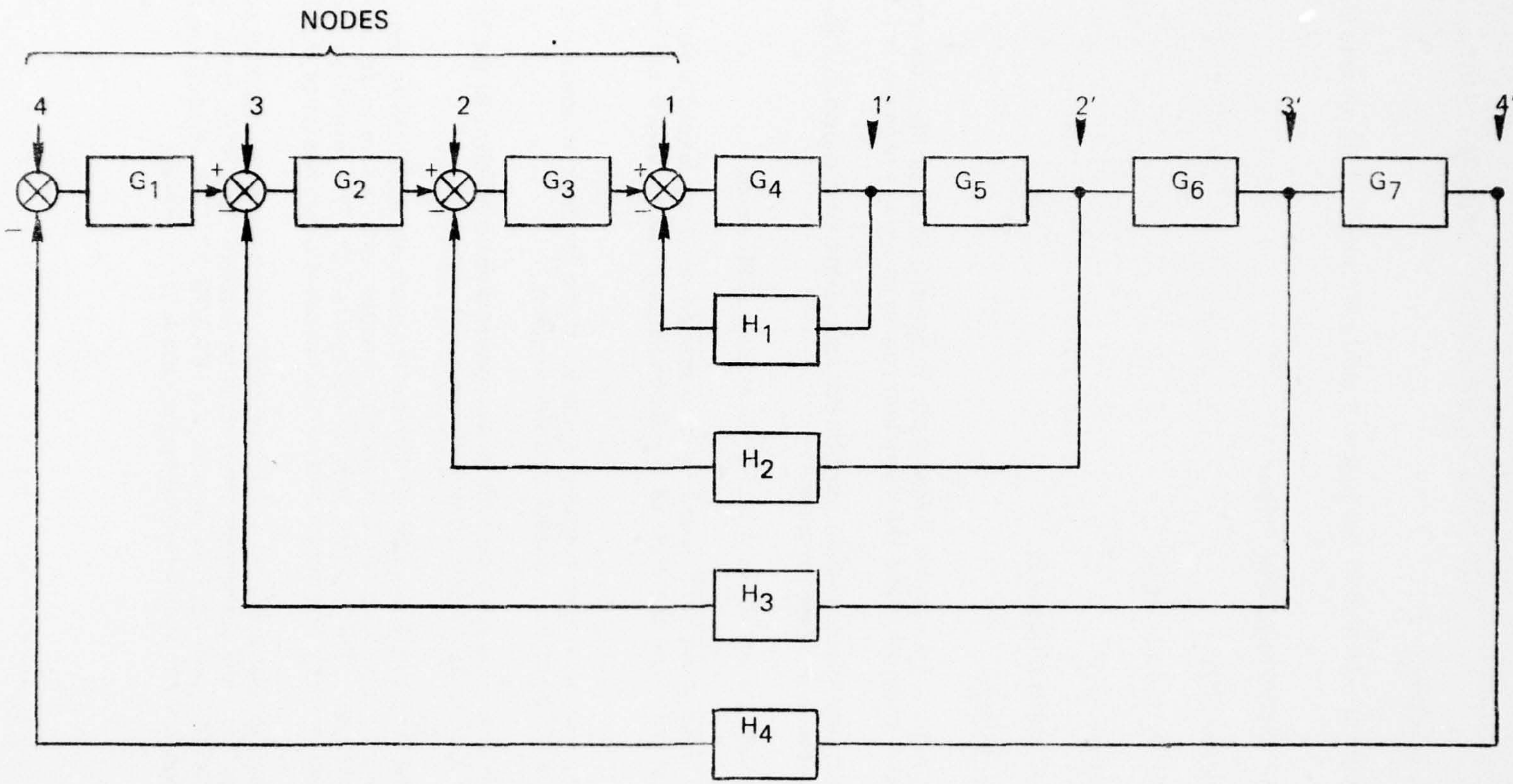

Figure 2. Generalized Block Diagram

h. The case identifier must not exceed ten alphanumeric characters.

i. When using either of the two versions of the stability analysis programs and when the execution is under the control of one of the three subroutines, halting the program and then continuing it will result in an error. This is due to the fact that the RETURN flag is lost when the program is halted. The way to recover from this error is to stop the execution and continue execution at some step in the main stability analysis program.

SECTION III

LOOP SOLVER PROGRAM (LSP)

1. PROGRAM DESCRIPTION

The loop solver program (LSP) has been designed to accept a complex block diagram with up to seven forward transfer functions and four feedback transfer functions, and to reduce them to an equivalent single forward transfer function and an associated equivalent feedback transfer function. The input data and the calculated equivalent transfer functions are stored on magnetic tape and are printed out. Data tape storage allows the transfer of the resulting polynomials to the tape input version of the SAP-TI.

The LSP is set up to process a block diagram of the form shown in Figure 2. Each of the transfer functions may consist of several polynomials which will be multiplied together as they are entered into the program. All of the individual transfer functions are initialized to zero, so it is only necessary to enter non-zero functions. The total system must not exceed 10th order.

The LSP will generate two equivalent transfer functions for each of the nodes shown in Figure 2. These two transfer functions are designated as (1) an internal loop equivalent and (2) an isolated loop equivalent. This terminology is not standard; thus, these terms are defined as follows:

Internal Loop Equivalent - The internal loop equivalent is the transfer function that is generated when all transfer functions between, or internal to, the designated node, e.g., 2, and its mirror node, e.g., 2′, are reduced to a single transfer function.

Isolated Loop Equivalent - The isolated loop equivalent is the transfer function that is generated when all transfer functions, both internal to and external to the designated node and its mirror node, are reduced to a single transfer function.

The two types of equivalents are illustrated for node 2 in Figure 3. As can be seen in Figure 3, the isolated loop equivalent is a combination of the internal loop equivalent plus an equivalent feedback, which is due to the external transfer functions. In general,

$$G]_x = G_{x-1} \left\{ \frac{G]_{x-1}}{1 + H_x G]_{x-1}} \right\} G]_{x+1}$$

where

$G]_x$ = the internal loop equivalent transfer function for node x

G_x = the forward transfer function numbered x

H_x = the feedback transfer function that is fed into node x

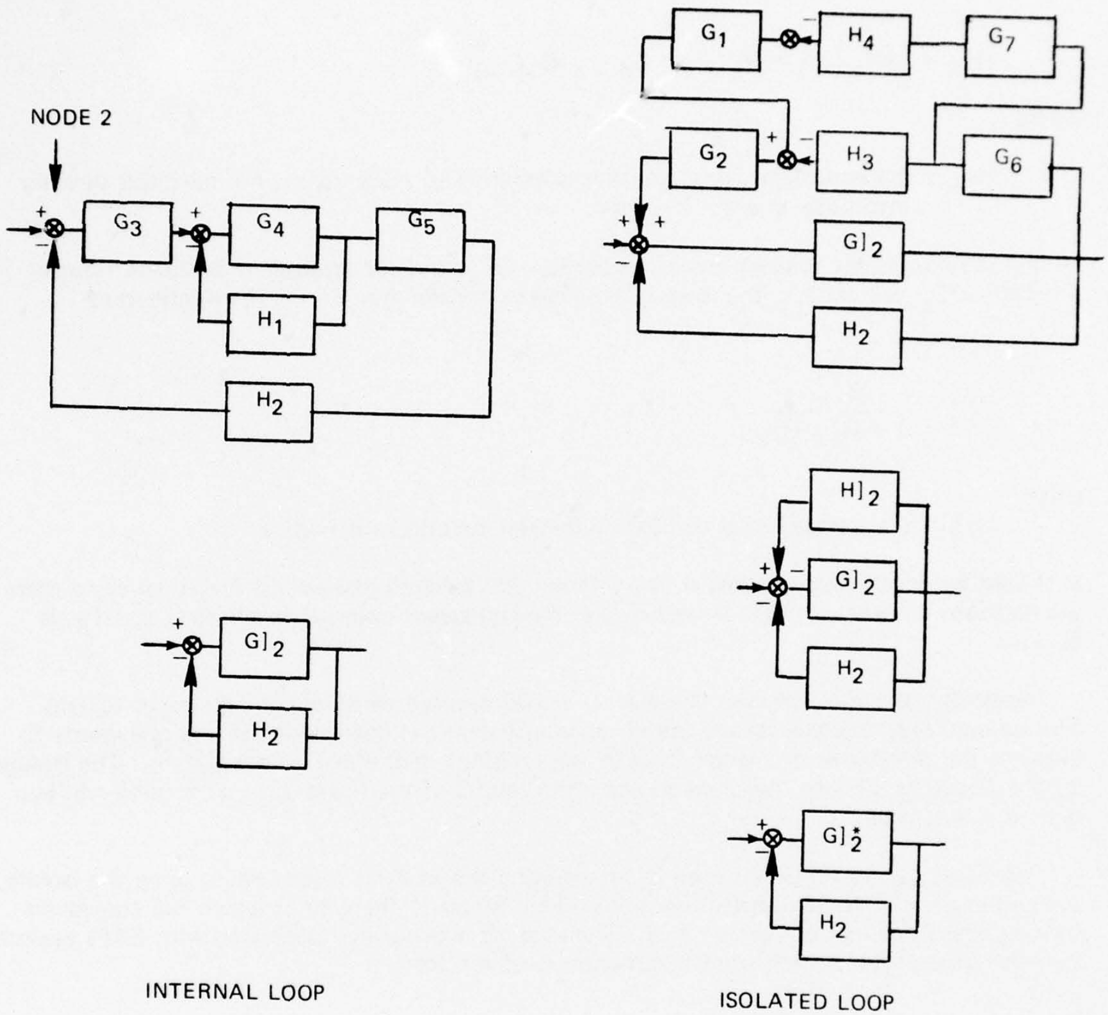

Figure 3. Equivalent Transfer Functions

The effect of the transfer functions external to node x can be considered as a single feedback and, in general, is

$$H]_x = H]_{x+1} + H_{x+1} \quad G_{4+x} G_{4-x}$$

where

$H]_x$ = the equivalent feedback that is external to node x, i.e., the feedback directly into node x is not included

Having the equivalent forward transfer function, $G]_x$, and the equivalent feedback transfer function, $H]_x$, for node x, the isolated equivalent transfer function is then defined as

$$G]_x^* = \frac{G]_x}{1 + G]_x H]_x}$$

where

$G]_x^*$ = the isolated loop equivalent transfer function for node x

It should be noted that, at node 4, the internal and isolated equivalents are equal since there are no loops external to node 4, and hence, the equivalent external feedback, i.e., $H]_4$, is zero.

Normally, the internal loop equivalents would be used when designing a servo system. The isolated loop equivalents are useful when a system has been designed and one wants to examine the sensitivity of a single loop in the presence of the rest of the system. The isolated loop equivalents are also useful when using the results of the linear analysis to verify full-up system simulation.

However, caution must be used in interpreting the analysis results when using the isolated loop equivalent forward transfer functions. This is due to the way in which the equivalent forward transfer functions are derived. The analysis subroutines associated with SAPs assume that the closed-loop system transfer function is of the form

$$\frac{C}{R} = \frac{KG}{1 + KGH}$$

The gain that is entered as a part of the analysis subroutines is the K in the above equation and is associated with the forward transfer function, G, or in this case, the equivalent forward transfer function, $G]_x$ or $G]_x^*$. The fact that the gain is associated with the isolated loop equivalent forward transfer function means that the gain is distributed in the individual forward transfer functions and in the internal feedback transfer functions, i.e., the gain cannot be associated with any one single transfer function in the original system. This limits the utility of the isolated loop equivalent in those cases where the closed-loop transfer function is used, e.g., the closed-loop Bode plot and the time response programs.

2. OPERATING PROCEDURE

a. Load the LSP into the HP 9830.

b. Insert a tape with a 2000-word data file in the calculator.

c. Press RUN/EXECUTE. The display will request the number of the forward transfer function that is to be entered. An entry of zero will cause the program to continue to the feedback transfer functions (step f). If a non-zero entry is made, the printer will print "ENTER NUMERATOR DATA FOR G# (entry)" and the display will request the order of the first polynomial in the numerator.

d. After the order is entered, the display will request the coefficients starting with the lowest order coefficient. A zero order polynomial with the constant equal to zero will terminate the input of the numerator.

e. The printer will print "ENTER DENOMINATOR DATA FOR G# (entry)", and the display will ask for the order of the first polynomial in the denominator. A zero order polynomial with the constant equal to zero will cause the program to return to step c.

f. The display will request the number of the feedback transfer function that is to be entered. An entry of zero will cause the program to continue to step g. A non-zero entry allows the entry of data, as in steps d and e.

g. The forward and feedback transfer function polynomials will be printed out in matrix format. The number of the column in each matrix corresponds to the number of the transfer function, as defined in Figure 2. The first row of the matrix is the zero order coefficient and so on up to the 10th order coefficient which is in row 11. The 12th row contains the order of the polynomial which is contained in that column.

h. The calculator will then calculate the equivalent loop transfer functions. This is time-consuming , and the calculator will appear to be totally inactive during this time.

i. The calculated internal and isolated loop equivalent transfer functions will be printed in the same format as in step g. The number of the columns corresponds to the number of the node, as defined in Figure 2.

j. The display will request the number of the data file in which the calculated data is to be stored. After the data is stored, the program will return to step c. This allows one to easily store a new set of equivalent loop data for a system which has only minor differences from the original one. Since the previous data for all the forward and feedback transfer functions are still in memory, it is only necessary to enter those which are different in the new system. If this option is not desired, merely press STOP.

3. EXAMPLE PROBLEM

The following system block diagram (Figure 4) can be put into the generalized format (Figure 2) in several ways. Two ways will be illustrated here:

Example 1:

Let $G_1 = 1$

$G_2 = (s + 1)/(s + 10)$

$G_3 = 1/s$

$G_4 = 1$

$G_5 = 0.5/(s + 10)$

$G_6 = 500/\pi s^2$

$G_7 = 1$

$H_1 = 0$

$H_2 = 0$

$H_3 = (s + 1)/1$

and

$H_4 = 0$

The sequence is which the data was entered is shown in Figure 5. The results are shown in Figure 6.

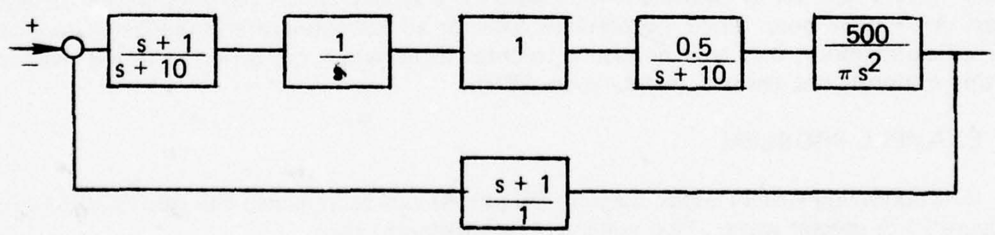

Figure 4. Example System

```
ENTER NUMERATOR DATA FOR G# 1
ENTER DENOMINATOR DATA FOR G# 1
ENTER NUMERATOR DATA FOR G# 2
ENTER DENOMINATOR DATA FOR G# 2
ENTER NUMERATOR DATA FOR G# 3
ENTER DENOMINATOR DATA FOR G# 3
ENTER NUMERATOR DATA FOR G# 4
ENTER DENOMINATOR DATA FOR G# 4
ENTER NUMERATOR DATA FOR G# 5
ENTER DENOMINATOR DATA FOR G# 5
ENTER NUMERATOR DATA FOR G# 6
ENTER DENOMINATOR DATA FOR G# 6
ENTER NUMERATOR DATA FOR G# 7
ENTER DENOMINATOR DATA FOR G# 7
ENTER NUMERATOR DATA FOR H# 3
ENTER DENOMINATOR DATA FOR H# 3
```

Figure 5. Input Sequence for LSP (Example 1)

FORWARD LOOP NUMERATORS

1.000E+00	1.000E+00	1.000E+00	1.000E+00	5.000E-01	5.000E+02	1.000E+00
0.000E+00	1.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	1.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

FORWARD LOOP DENOMINATORS

1.000E+00	1.000E+01	0.000E+00	1.000E+00	1.000E+01	0.000E+00	1.000E+00
0.000E+00	1.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	0.000E+00
0.000E+00	2.000E+00	0.000E+00	0.000E+00	0.000E+00	3.142E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	1.000E+00	1.000E+00	0.000E+00	1.000E+00	2.000E+00	0.000E+00

FEEDBACK LOOP NUMERATORS

0.000E+00	0.000E+00	1.000E+00	0.000E+00
0.000E+00	0.000E+00	1.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	1.000E+00	0.000E+00

FEEDBACK LOOP DENOMINATORS

1.000E+00	1.000E+00	1.000E+00	1.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00

Figure 6. LSP Output (Example 1)

```
INTERNAL LOOP EQUIVALENT FORWARD NUMERATORS
 1.000E+00  5.000E-01  2.500E+02  2.500E+02
 0.000E+00  0.000E+00  2.500E+02  2.500E+02
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  1.000E+00  1.000E+00

INTERNAL LOOP EQUIVALENT FORWARD DENOMINATORS
 1.000E+00  0.000E+00  0.000E+00  2.500E+02
 0.000E+00  1.000E+01  0.000E+00  5.000E+02
 0.000E+00  1.000E+00  0.000E+00  2.500E+02
 0.000E+00  0.000E+00  3.142E+02  3.142E+02
 0.000E+00  0.000E+00  6.283E+01  6.283E+01
 0.000E+00  0.000E+00  3.142E+00  3.142E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  2.000E+00  5.000E+00  5.000E+00

ISOLATED LOOP EQUIVALENT FORWARD NUMERATORS
 0.000E+00  0.000E+00  2.500E+02  2.500E+02
 0.000E+00  0.000E+00  2.500E+02  2.500E+02
 0.000E+00  1.571E+01  0.000E+00  0.000E+00
 3.142E+02  1.571E+00  0.000E+00  0.000E+00
 6.283E+01  0.000E+00  0.000E+00  0.000E+00
 3.142E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 5.000E+00  3.000E+00  1.000E+00  1.000E+00

ISOLATED LOOP EQUIVALENT FORWARD DENOMINATORS
 2.500E+02  2.500E+02  0.000E+00  2.500E+02
 5.000E+02  5.000E+02  0.000E+00  5.000E+02
 2.500E+02  2.500E+02  0.000E+00  2.500E+02
 3.142E+02  3.142E+02  3.142E+02  3.142E+02
 6.283E+01  6.283E+01  6.283E+01  6.283E+01
 3.142E+00  3.142E+00  3.142E+00  3.142E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 5.000E+00  5.000E+00  5.000E+00  5.000E+00
```

Figure 6. LSP Output (Example 1) (Concluded)

Example 2:

$$\text{Let } G_4 = \frac{(s + 1)\ (0.5)\ (500)}{(s + 10)\ (s)\ (s + 10)\ (s^2)\,\pi}$$

and

$$H_1 = (s + 1)/1$$

The sequence in which the data was entered is shown in Figure 7, and the results are shown in Figure 8.

It can be seen that the data for node 1 in example 2 is exactly equal to the data for node 3 in example 1.

4. FLOW CHART

The flow chart for the LSP is shown in Figure 9. The numbers in parantheses on the right and left side of the chart are the line numbers that should be used to initiate that particular function or process in the flow diagram. The complete listing for the program is contained in Appendix A.

```
ENTER NUMERATOR DATA FOR G# 4

ENTER DENOMINATOR DATA FOR G# 4

ENTER NUMERATOR DATA FOR H# 1

ENTER DENOMINATOR DATA FOR H# 1
```

Figure 7. Input Sequence for LSP (Example 2

```
FORWARD LOOP NUMERATORS
 0.000E+00  0.000E+00  0.000E+00  2.500E+02  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  2.500E+02  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  1.000E+00  0.000E+00  0.000E+00  0.000E+00

FORWARD LOOP DENOMINATORS
 1.000E+00  1.000E+00  1.000E+00  0.000E+00  1.000E+00  1.000E+00  1.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  3.142E+02  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  6.283E+01  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  3.142E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  5.000E+00  0.000E+00  0.000E+00  0.000E+00

FEEDBACK LOOP NUMERATORS
 1.000E+00  0.000E+00  0.000E+00  0.000E+00
 1.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 1.000E+00  0.000E+00  0.000E+00  0.000E+00

FEEDBACK LOOP DENOMINATORS
 1.000E+00  1.000E+00  1.000E+00  1.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
```

Figure 8. LSP Output (Example 2)

```
INTERNAL LOOP EQUIVALENT FORWARD NUMERATORS
 2.500E+02  0.000E+00  0.000E+00  0.000E+00
 2.500E+02  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 1.000E+00  0.000E+00  0.000E+00  0.000E+00

INTERNAL LOOP EQUIVALENT FORWARD DENOMINATORS
 0.000E+00  1.000E+00  1.000E+00  1.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 3.142E+02  0.000E+00  0.000E+00  0.000E+00
 6.283E+01  0.000E+00  0.000E+00  0.000E+00
 3.142E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 5.000E+00  0.000E+00  0.000E+00  0.000E+00

ISOLATED LOOP EQUIVALENT FORWARD NUMERATORS
 2.500E+02  0.000E+00  0.000E+00  0.000E+00
 2.500E+02  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 1.000E+00  0.000E+00  0.000E+00  0.000E+00

ISOLATED LOOP EQUIVALENT FORWARD DENOMINATORS
 0.000E+00  1.000E+00  1.000E+00  1.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 3.142E+02  0.000E+00  0.000E+00  0.000E+00
 6.283E+01  0.000E+00  0.000E+00  0.000E+00
 3.142E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 0.000E+00  0.000E+00  0.000E+00  0.000E+00
 5.000E+00  0.000E+00  0.000E+00  0.000E+00
```

Figure 8. LSP Output (Example 2) (Concluded)

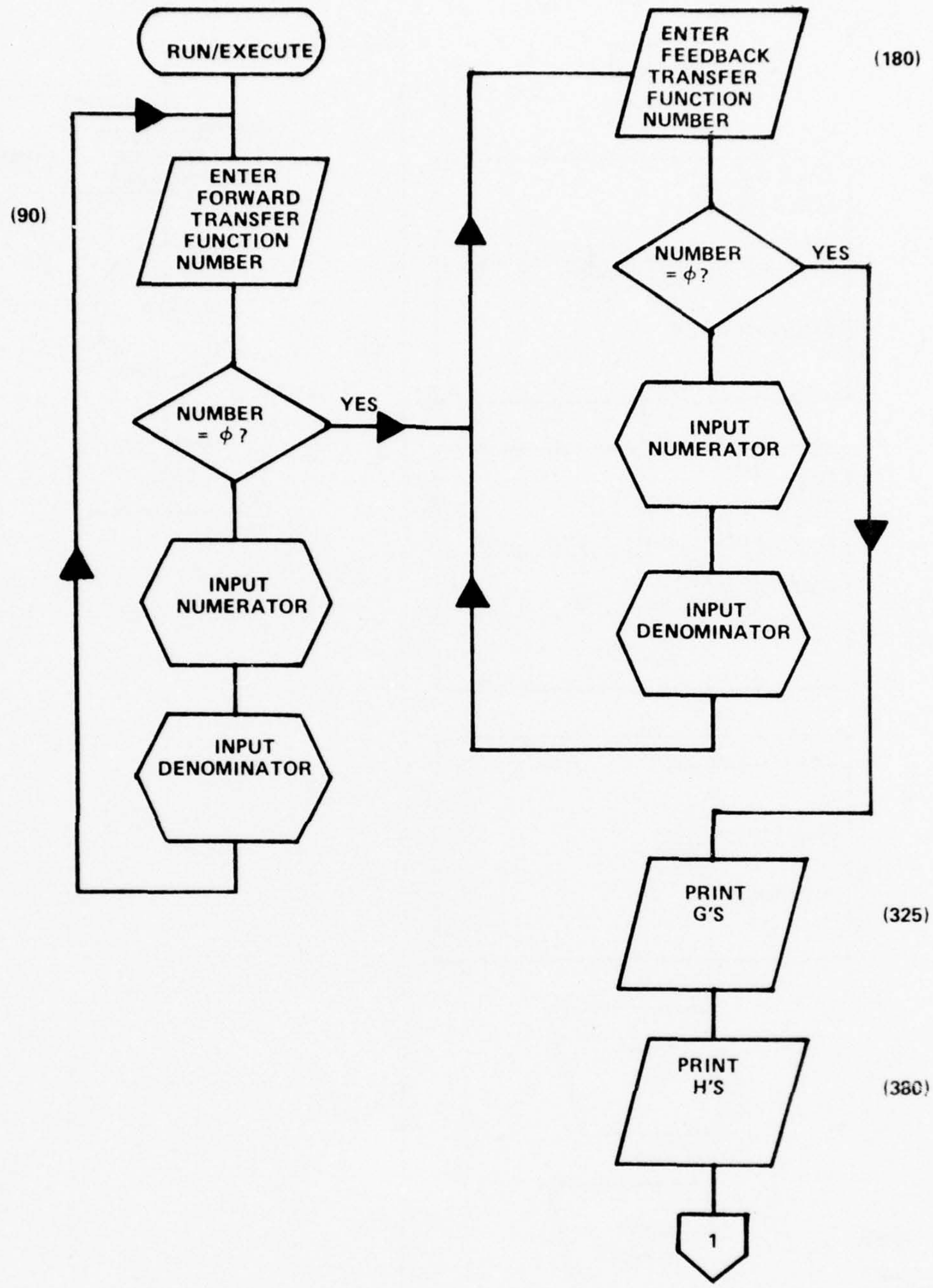

Figure 9. LSP Flow Chart

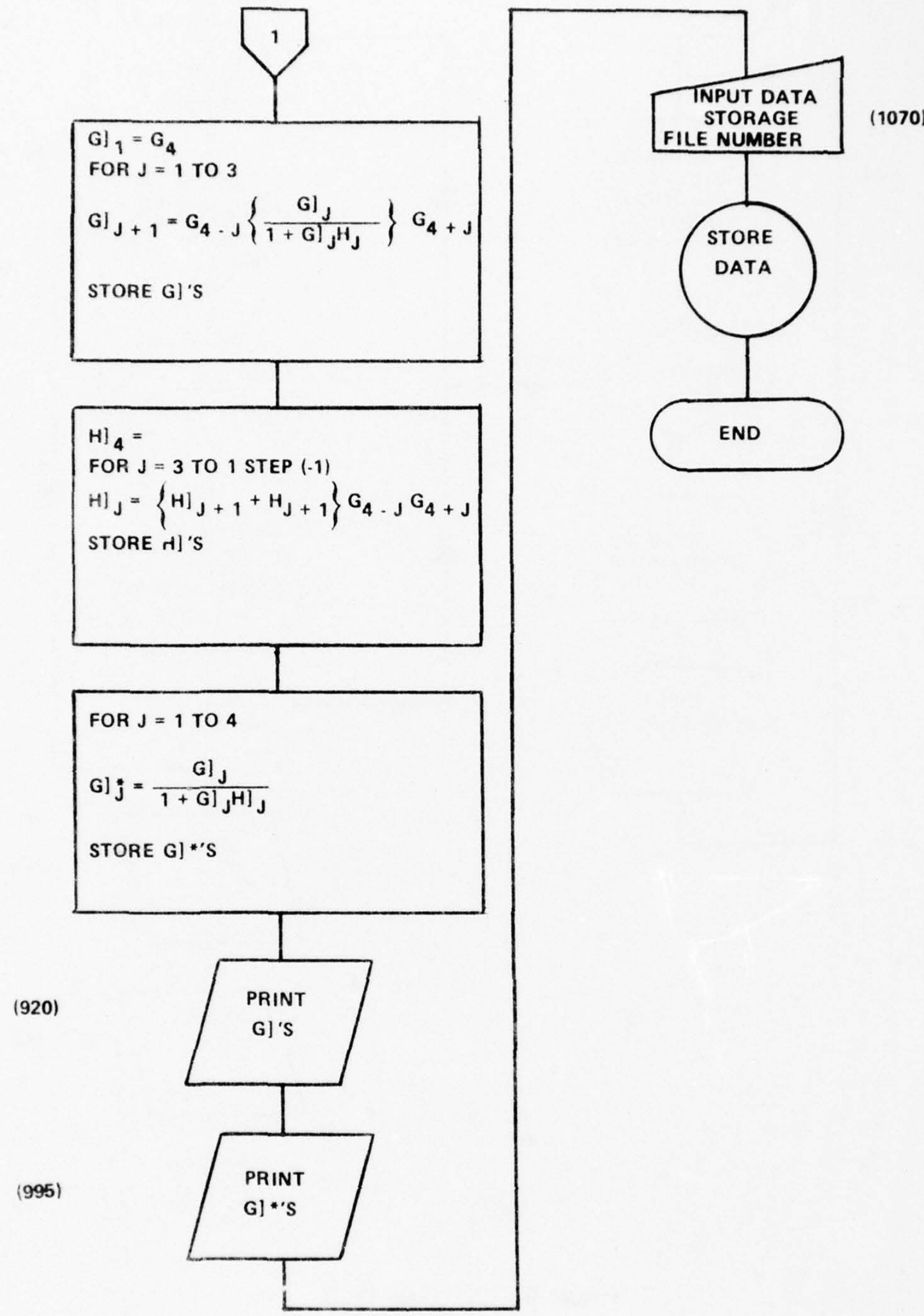

Figure 9. LSP Flow Chart (Concluded)

SECTION IV

STABILITY ANALYSIS PROGRAM-TAPE INPUT (SAP-TI)

1. PROGRAM DESCRIPTION

The SAP-TI has been designed to retrieve designated transfer functions from the data tape generated by the LSP and to control the execution of the three analysis subroutines, i.e., the Root Locus (RL), the Bode (B), and the Time Response (TR) subroutines.

The program will load common with data from the user specified data file and then retrieve the equivalent forward and feedback transfer functions for the node and loop type which are specified by the user. The program then constructs three polynomials which are used in the three subroutines. The three polynomials are as follows:

NG*DH - (Numerator of G)*(Denominator of H)

NG*NH = (Numerator of G)*(Numerator of H)

DG*DH = (Denominator of G)*(Denominator of H)

Both the characteristic equation and the closed loop transfer function can be obtained from these three polynomials since

$$\frac{C}{R} = \frac{KG}{1 + KGH} = \frac{K(NG)\,(DH)}{(DG)(DH) + K(NG)(NH)}$$

The program then loads the appropriate subroutine and transfers control to the subroutine.

2. OPERATING PROCEDURE

a. Load the SAP-TI into the HP 9830.

b. Insert the data tape in the calculator.

c. Press RUN/EXECUTE. The display will request the number of the data file. The data will be loaded into common from the designated data file.

d. The display will request the loop type, i.e., internal or isolated. The definitions of the internal and isolated loops are the same as in the LSP description.

e. The display will request the node number which is defined in Figure 2.

f. The printer will print the statement "ROOT LOCUS (L), BODE (B), RESPONSE (R), OR STOP (S)?". There will be a short time delay between steps e and f because the calculator is generating the three polynomials referred to in the program description. Before entering the proper alphanumeric character in response to the printed question, be sure that the program tape is replaced in the calculator.

g. The calculator will load the selected subroutine into memory. There are internal flags set in the program so that this step will be skipped if the selected subroutine is already in memory.

h. At this point, the selected subroutine will be executed. The operating procedures for the subroutines are contained in the following sections of this report.

i. After the subroutine is executed, the display will ask if a new transfer function is desired. If the response is negative, the program will return to step f.

j. If a new transfer function is requested, the display will ask whether the new transfer function is in the existing data file or in a new data file. If the existing file is indicated, the program will return to step d.

k. If the new transfer function is in a new data file, put the data tape which contains the appropriate data file in the calculator. The display will request the number of the data file, the data will be loaded into common, and the program will return to step d.

3. FLOW CHART

The flow chart for the SAP-TI program is shown in Figure 10. The numbers in parantheses on the right and left sides of the chart are the line numbers that should be used to continue program execution at that particular function or process on the flow diagram. The complete listing for the program is contained in Appendix B.

Two flags are used to keep track of which of the three subroutines are in memory. Two flags, one in common and one in memory, are used because loading a new data file will destroy the flag that is in common and the loading of a new subroutine will destroy the flag that is not in common. The use of the two flags insures that there is always one flag to indicate which subroutine is in memory.

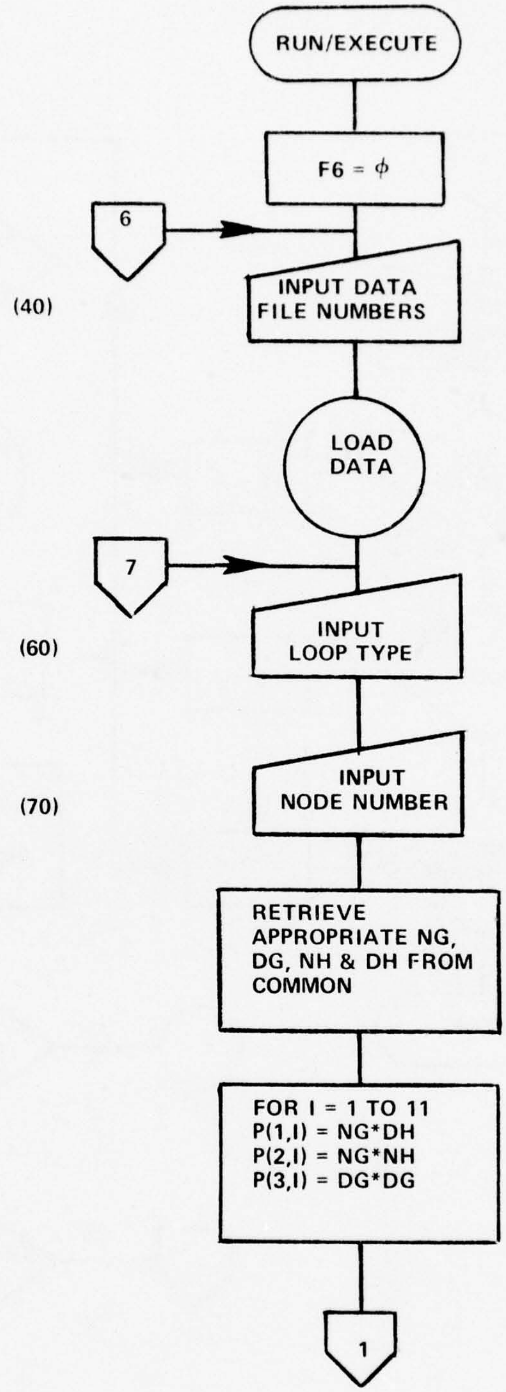

Figure 10. SAP-TI Flow Chart

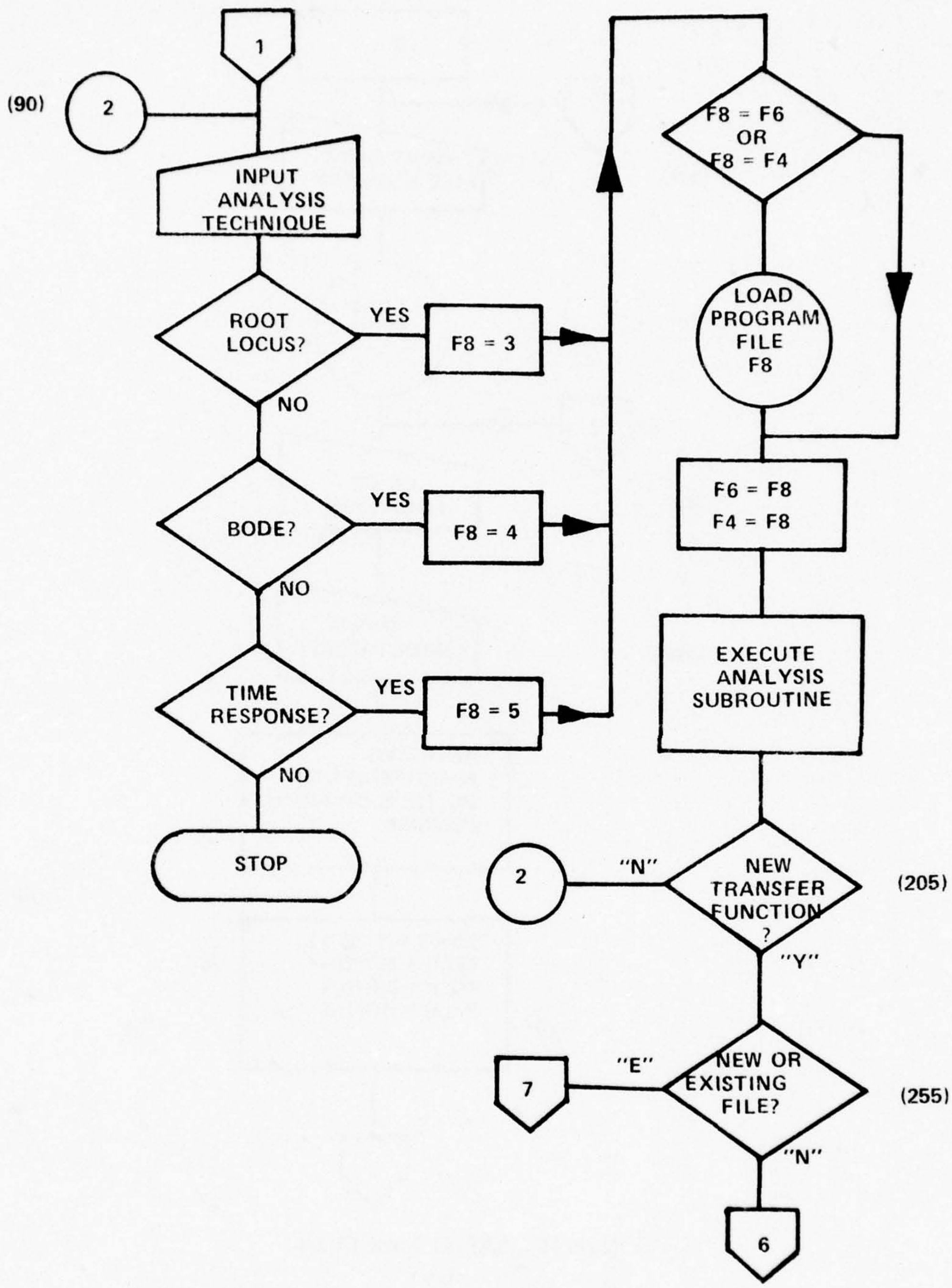

Figure 10. SAP-TI Flow Chart (Concluded)

SECTION V

STABILITY ANALYSIS PROGRAM-KEYBOARD INPUT (SAP-KI)

1. PROGRAM DESCRIPTION

The SAP-KI is a combination of the LSP and the SAP-TI programs. The program allows the user to enter the forward and feedback functions directly into the stability analysis program without having to use the LSP and data tape. This program is useful when the system is simple enough that the block diagram reduction can be done manually.

2. OPERATING PROCEDURE

a. Load the SAP-KI into the HP 9830.

b. Press RUN/EXECUTE. The printer will print "ENTER NUMERATOR OF G", and the display will request the order of the first polynomial in the numerator.

c. After the order is entered, the display will request the coefficients starting with the lowest order coefficient. A zero order polynomial with the numerator.

d. The printer will print "ENTER DENOMINATOR OF G", and the display will request the order of the polynomial as in step b and the coefficients are entered as in step c.

e. The numerator of H is entered as in steps b and c.

f. The denominator of H is entered as in steps b and c.

g. The printer will print the statement "ROOT LOCUS (L), BODE (B), RESPONSE (R), OR STOP (S)?". There will be a short time delay between steps f and g because the calculator is generating the three polynomials referred to in the SAP-TI Program Description in Section III.

h. The calculator will load the selected subroutine into memory. There are internal flags set in the program so that this step will be skipped if the selected subroutine is already in memory.

i. At this point, the selected subroutine will be executed. The operating procedures for the subroutines are contained in the following sections of this report.

j. After the subroutine is executed, the display will ask if a new transfer function is desired. If the response is negative, the program will return to step g.

j. If a new transfer function is requested, the program will return to step b.

3. EXAMPLE PROBLEM

Figure 11 shows the keyboard input sequence for the system shown in Figure 4. For this example the PRINT ALL option on the calculator was used to show the information that is

```
ENTER NUMERATOR OF G

ORDER OF POLYNOMIAL=?1
A( 0      )=?1
A( 1      )=?1
ORDER OF POLYNOMIAL=?0
CONSTANT=?.5
ORDER OF POLYNOMIAL=?0
CONSTANT=?500
ORDER OF POLYNOMIAL=?0
CONSTANT=?0
ENTER DENOMINATOR OF G

ORDER OF POLYNOMIAL=?1
A( 0      )=?10
A( 1      )=?1
ORDER OF POLYNOMIAL=?1
A( 0      )=?0
A( 1      )=?1
ORDER OF POLYNOMIAL=?1
A( 0      )=?10
A( 1      )=?1
ORDER OF POLYNOMIAL=?2
A( 0      )=?0
A( 1      )=?0
A( 2      )=?3.14159
ORDER OF POLYNOMIAL=?0
CONSTANT=?0
ENTER NUMERATOR OF H

ORDER OF POLYNOMIAL=?1
A( 0      )=?1
A( 1      )=?1
ORDER OF POLYNOMIAL=?0
CONSTANT=?0
ENTER DENOMINATOR OF H

ORDER OF POLYNOMIAL=?0
CONSTANT=?1
ORDER OF POLYNOMIAL=?0
CONSTANT=?0
```

Figure 11. Input Sequence for SAP-KI

normally on the display. In normal operation only the four statements that are designated by the arrow would appear on the printer. The questions will normally appear on the display, and the number following the question mark is the number that the user inputs in response to that question.

4. FLOW CHART

The flow chart for the SAP-KI program is shown in Figure 12. The numbers in parantheses on the right and left sides of the chart are the line numbers that should be used to continue the program execution at that particular function or process on the flow diagram. The complete listing for the program is contained in Appendix C. F4 is a flag to indicate what subroutine is in memory.

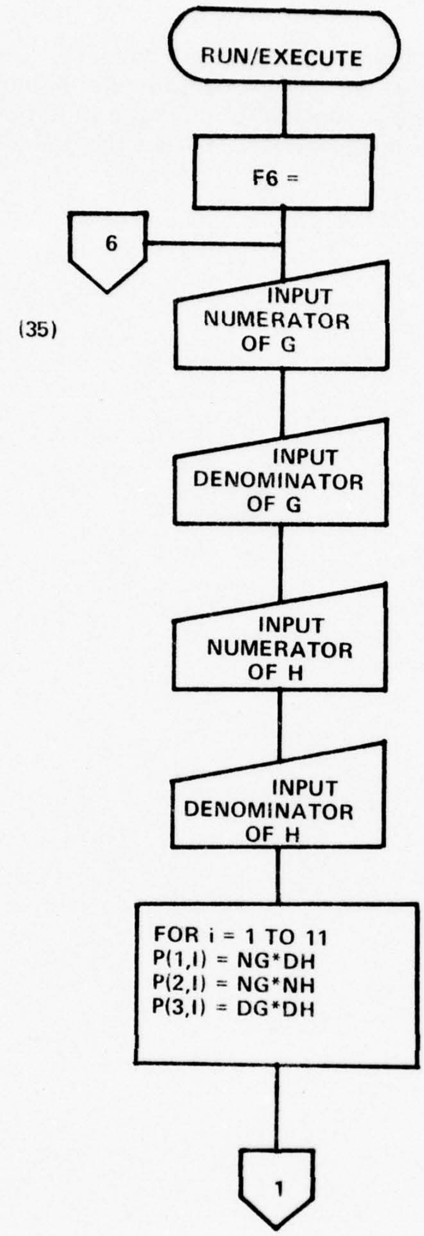

Figure 12. SAP-KI Flow Chart

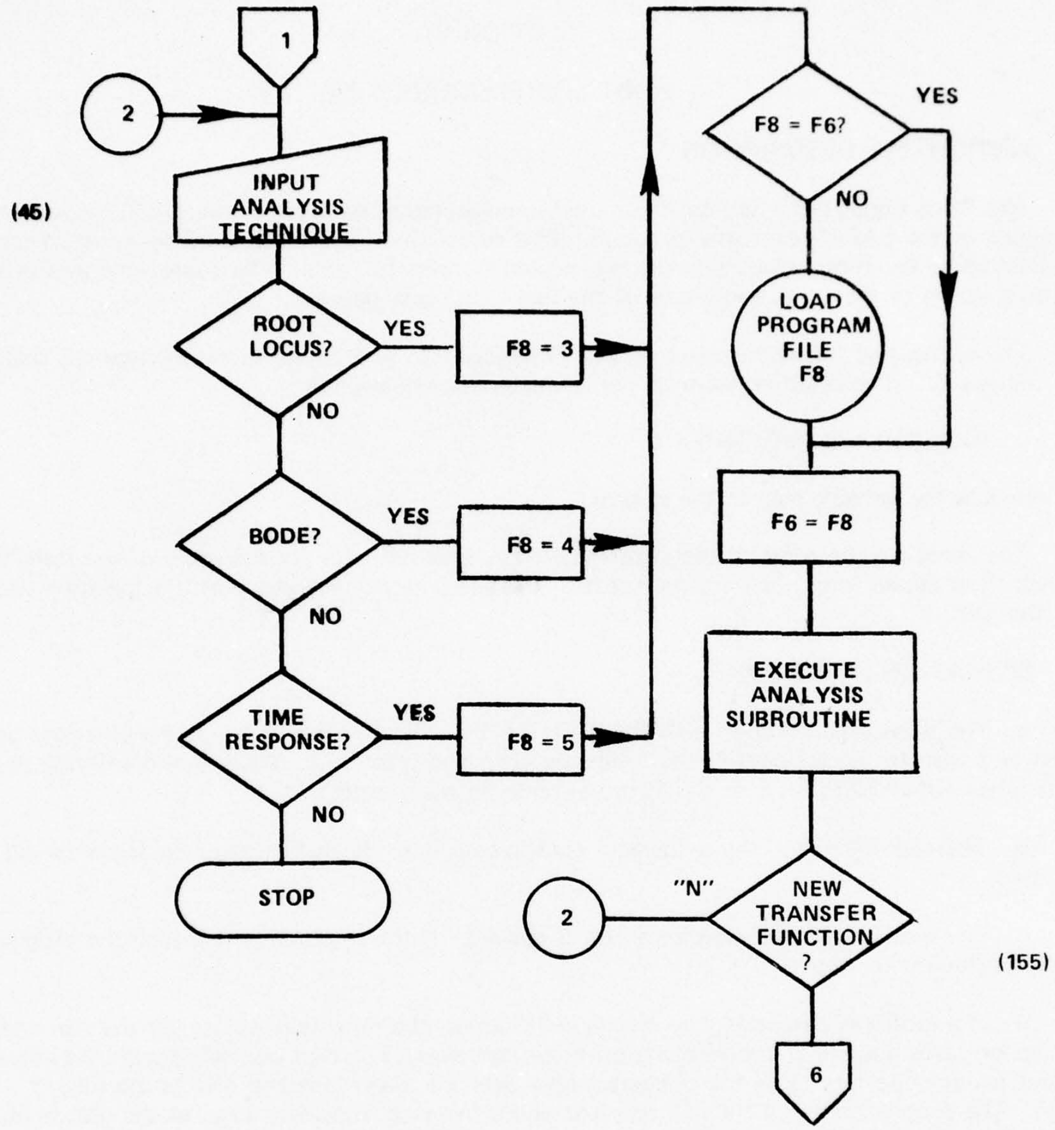

Figure 12. SAP-KI Flow Chart (Concluded)

SECTION VI

ROOT LOCUS SUBROUTINE

1. SUBROUTINE DESCRIPTION

The Root Locus (RL) Subroutine is used in conjunction with either the SAP-TI executive program or the SAP-KI executive program. This subroutine finds the loci of the roots of the characteristic equation associated with the system transfer function. The subroutine also provides a listing of the roots and a plot of the loci if the user desires.

The subroutine forms the characteristic equation with two of the three polynomials referenced in Section IV. The resulting form of the characteristic equation is

$$(DG)(DH) + K\,(NG)(NH) = 0$$

where K is the variable gain in the system.

The zeros are the roots of the term (NG)(NH), and the poles are the roots of the term (DG)(DH). The closed loop poles are the roots of the characteristic equation with a specified value of the gain, K.

2. OPERATING PROCEDURE

a. The display will request a CASE IDENTIFIER. Type in up to ten alphanumeric characters that you wish to use to identify the following plots and print-outs. After the identifying characters, press either EXECUTE or STOP to continue program execution.

b. The case identifier, the numerator coefficients, and the denominator coefficients will be printed.

c. The display will ask whether a plot is desired. If the response is negative, the program will continue with step f.

d. If a plot was requested, the display will request the maximum values for the real and imaginary axes and for the increments on these two axes. The sign associated with the values input is not important since the calculator only uses the magnitude and affixes the proper sign. The program draws a six-inch real axis and a five-inch imaginary axis, so the plot is more readable if the maximum value of the real axis is a multiple of six and the maximum imaginary value is a multiple of five. Prior to entering the imaginary increment, be sure that the plotter is set up properly.

e. The axes will be drawn and labeled. It should be noted that the program only plots one quadrant of the S-plane.

f. The zeros will be calculated and printed (and plotted). The first number printed is the real part of the root, and the second number is the imaginary part of the root.

g. The poles will be calculated and printed (and plotted).

h. The display will request the minimum gain, maximum gain, and gain increment. The gain is the value of K referenced in the Program Description, above.

i. The roots of the characteristic equation for each value of gain which was specified in step h will be calculated and printed (and plotted).

j. The display will ask if a new gain increment is desired. If the response is affirmative, the program will return to step h. If the response is negative and if a plot was not originally requested, the program will return control to the SAP executive program.

k. If the response is negative and if a plot was originally requested, the display will ask if a replot is desired. This option is provided in case the user wants to expand some portion of the original plot. If a plot was originally requested, the coordinates of each point that was plotted were also stored in a storage array. Due to the dimensions of the storage array, only the first 65 points are stored. When a replot is requested, the coordinates of each previously stored point are retrieved from the storage array, examined to see if they are within the new scale, and if so, the point is plotted. If the response to the request for a replot is negative, the program control will be returned to the SAP executive program. If the response is affirmative, the display will request information about the replot axes as in steps d and e, above. Prior to entering the value of the imaginary increment, be sure that the plotter is set up.

1. The previously calculated roots will be replotted; they will not be recalculated. The program will then return to step k, above.

3. EXAMPLE PROBLEM

The system in Figure 4 is used in this example. The SAP-KI executive program is used. The printer output is shown in Figure 13. The option for a NEW GAIN INCREMENT was used. The original gain increment was from 0.1 to 1 in steps of 0.1. The additional gain increment was from 2 to 16 in steps of 2. The results of this option can be seen in the printout and in the spacing of the roots on the plot. The original plot is shown in Figure 14, and a sample replot is shown in Figure 15. Note the difference in the value of the axes in the two plots. The replot greatly expands the movement of the roots in the vicinity of the origin.

4. FLOW CHART

The flow chart for the RL subroutine is shown in Figure 16. The numbers in parantheses on the right and left of the chart are the line numbers that should be used to initiate that particular function or process in the flow diagram. The complete listing for the subroutine is contained in Appendix D.

```
ROOT LOCUS(L),BODE(B),RESPONSE(T) OR STOP(S)?

ROOT LOCUS SELECTED

***************
CASE EXAMPLE
***************
NUMERATOR COEFFICIENTS
N( 0      )= 250
N( 1      )= 500
N( 2      )= 250
DENOMINATOR COEFFICIENTS
D( 0      )= 0
D( 1      )= 0
D( 2      )= 0
D( 3      )= 314.159
D( 4      )= 62.8318
D( 5      )= 3.14159

ZEROS
-1.00              0.00
-1.00              0.00

POLES
 0.00              0.00
 0.00              0.00
 0.00              0.00
-10.00             0.00
-10.00             0.00


GAIN= 0.10
-0.34              0.00
 0.14              0.47
 0.14             -0.47
-9.17              0.00
-10.78             0.00

GAIN= 0.20
-0.40              0.00
 0.15              0.62
 0.15             -0.62
-8.81              0.00
-11.09             0.00

GAIN= 0.30
-0.44              0.00
 0.14              0.74
 0.14             -0.74
-8.52              0.00
-11.32             0.00
```

Figure 13. RL Subroutine Output

```
GAIN= 0.40
-0.46        0.00
 0.13        0.84
 0.13       -0.84
-8.28        0.00
-11.52       0.00

GAIN= 0.50
-0.49        0.00
 0.11        0.92
 0.11       -0.92
-8.05        0.00
-11.69       0.00

GAIN= 0.60
-0.51        0.00
 0.10        1.00
 0.10       -1.00
-7.85        0.00
-11.84       0.00

GAIN= 0.70
-0.52        0.00
 0.08        1.08
 0.08       -1.08
-7.65        0.00
-11.98       0.00

GAIN= 0.80
-0.54        0.00
 0.06        1.15
 0.06       -1.15
-7.47        0.00
-12.10       0.00

GAIN= 0.90
-0.55        0.00
 0.03        1.21
 0.03       -1.21
-7.30        0.00
-12.22       0.00

GAIN= 1.00
-0.56        0.00
 0.01        1.27
 0.01       -1.27
-7.13        0.00
-12.35       0.00

GAIN= 2.00
-0.63        0.00
-0.27        1.83
-0.27       -1.83
-5.63        0.00
-13.21       0.00

GAIN= 4.00
-0.70        0.00
-3.14        0.00
-0.89        3.05
-0.89       -3.05
-14.38       0.00

GAIN= 6.00
-0.73        0.00
-2.10        0.00
-0.97        4.41
-0.97       -4.41
-15.23       0.00

GAIN= 8.00
-0.76        0.00
-1.77        0.00
-0.78        5.41
-0.78       -5.41
-15.93       0.00

GAIN= 10.00
-0.78        0.00
-1.61        0.00
-0.55        6.19
-0.55       -6.19
-16.52       0.00

GAIN= 12.00
-0.79        0.00
-1.51        0.00
-0.33        6.84
-0.33       -6.84
-17.05       0.00

GAIN= 14.00
-0.80        0.00
-1.45        0.00
-0.12        7.41
-0.12       -7.41
-17.52       0.00

GAIN= 16.00
-0.81        0.00
-1.40        0.00
 0.08        7.90
 0.08       -7.90
-17.95       0.00
```

Figure 13. RL Subroutine Output (Concluded)

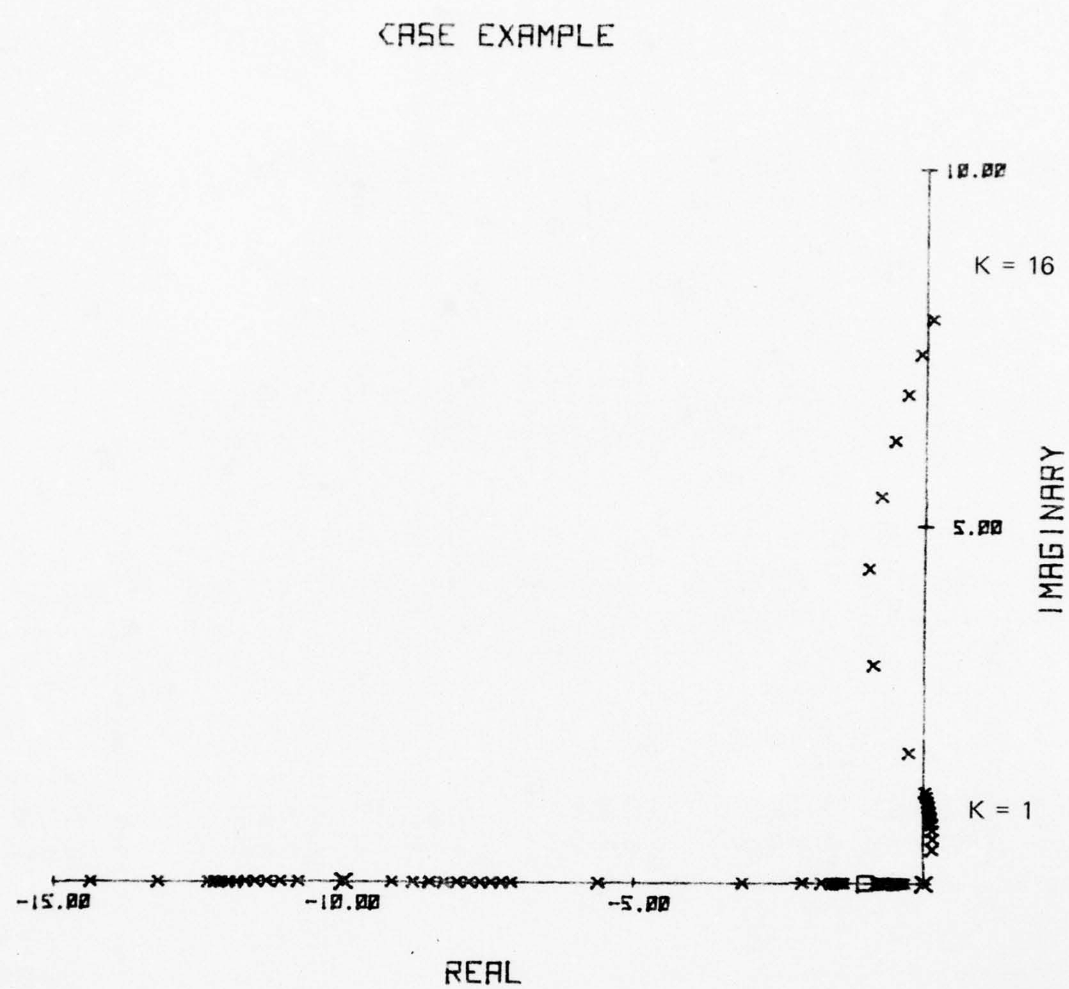

Figure 14. RL Subroutine Plot

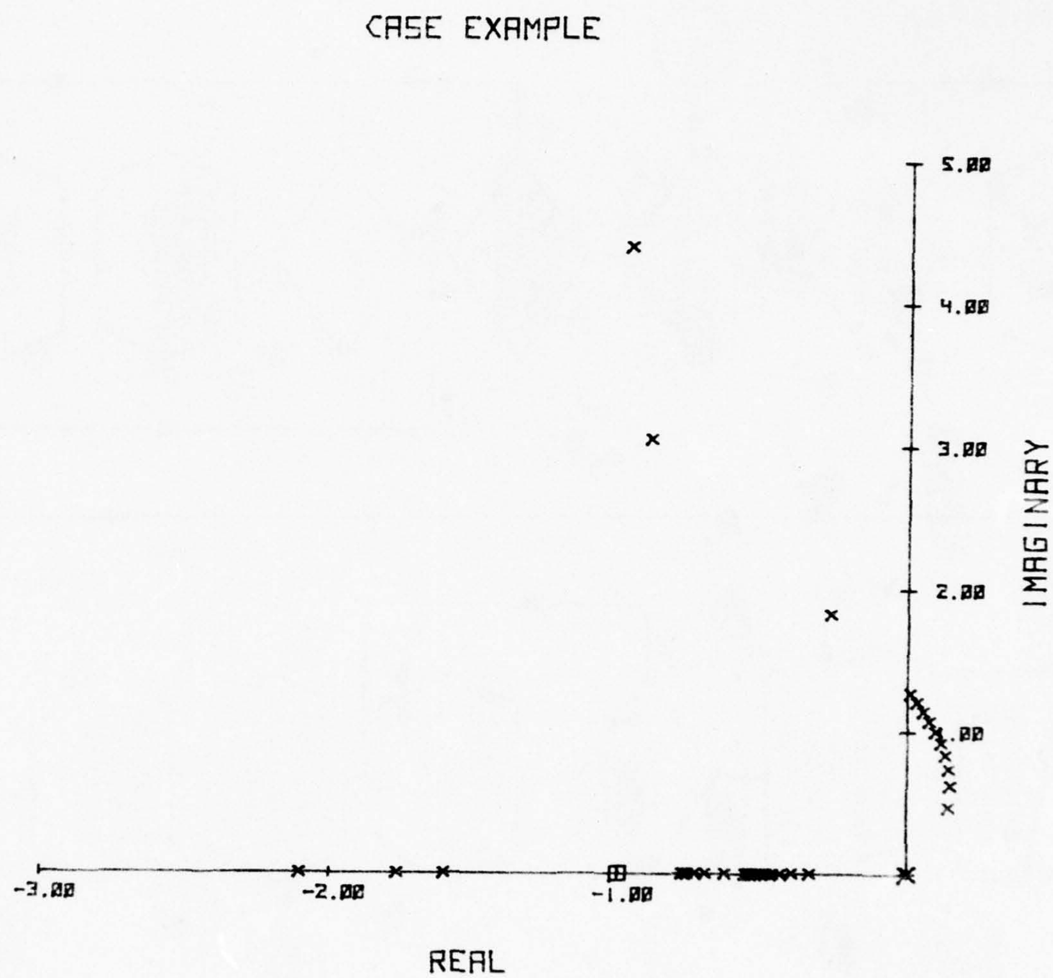

Figure 15. RL Subroutine Replot

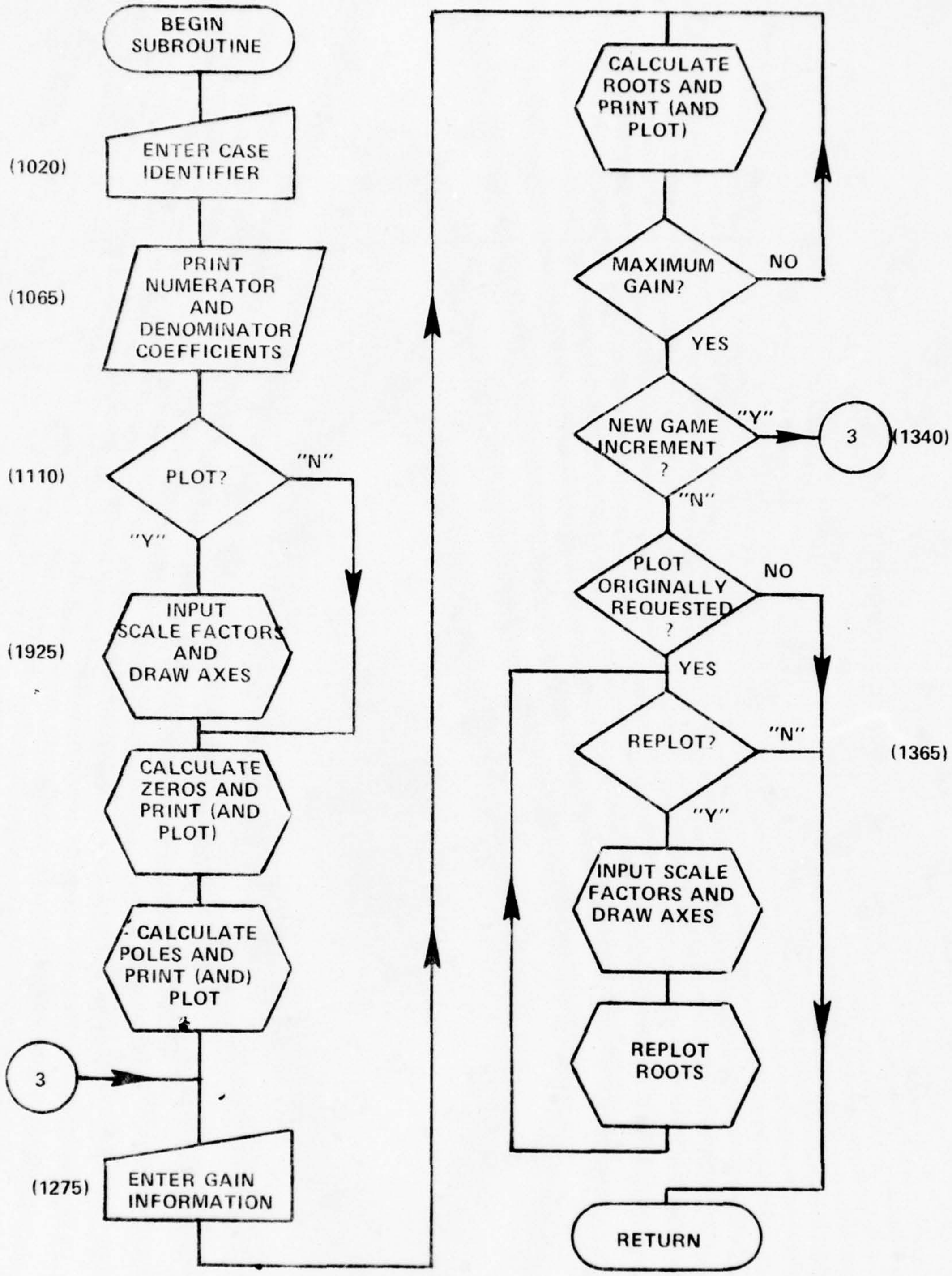

Figure 16. RL Subroutine Flow Chart

SECTION VII

BODE SUBROUTINE

1. PROGRAM DESCRIPTION

The Bode (B) Subroutine is used in conjunction with either the SAP-TI executive program or the SAP-KI executive program. The subroutine will calculate and plot either the Bode stability criteria or the Bode frequency response.

The Bode stability criteria, which is referred to as the "open loop" response in this program, are based on the term (GH) of the characteristic equation. A plot of the amplitude and phase of the term (GH) will yield the gain and phase margins of the system. In this portion of the program, the gain is taken as unity. A gain other than one will simply translate the amplitude curve in the Y direction. The phase margin is defined as

$$PM = 180 - |\text{phase}| \text{ evaluated at that frequency}$$

where the amplitude is one.

The gain margin is defined as

$$GM = |\text{amplitude}| \text{ evaluated at that frequency}$$

where the phase is 180 degrees .

The Bode frequency response which is referred to as the "closed loop" response in this program is based on the closed loop transfer function

$$\frac{C}{R} = \frac{KG}{1 + K\,(GH)}$$

In this case, the response of the system is dependent on the gain of the system, and, therefore, a provision for entering a gain is incorporated in the program. As implied by the sign in the denominator of the above equation, the program assumes negative feedback. A system which has positive feedback can be evaluated by affixing a negative sign to the feedback transfer function.

2. OPERATING PROCEDURE

a. The display will request a CASE IDENTIFIER. Type in up to ten alphanumeric characters that you wish to use to identify the following plots and print-outs. After the identifying characters are entered, press either EXECUTE or STOP to continue program execution.

b. The case identifier will be printed, and the display will ask whether the open or closed loop version is desired. If the response is open loop, the program will proceed to step d.

c. If the response is closed loop, the display will request the desired gain.

d. The numerator coefficients and the denominator coefficients will be printed, and the display will request the frequency range over which the response is to be evaluated. The range is in powers of ten. For example, if the desired frequency range is 0.1 rad/sec to 100 rad/sec, the input to the program would be -1, 2 since $0.1 = 10^{-1}$ and $100 = 10^2$.

e. The calculator will then calculate the amplitude and phases for the specified frequency range. This is time-consuming, and the calculator will appear to be totally inactive for several minutes.

f. The display will request whether a listing of the phase and amplitude is desired. A listing will or will not be printed, depending on the response. The program will automatically scale the amplitude plot up to a maximum of 120 dB. The phase is scaled from +180 degrees to -180 degrees. The relative magnitude and the quadrant of the phase angle are correct, but the absolute magnitude of the phase angle will have to be determined by the user. For example in the sample problem, the phase angle for very low frequencies should be approximately -270 degrees, i.e., two lead terms and five lag terms. The program will only determine that the phase angle due to the five lags is approximately 90 degrees and is in the first quadrant. The result of this is that the total phase angle at low frequencies will be calculated as +90 degrees instead of the correct -270 degrees.

g. If the open loop response was originally requested, the gain and phase margins will be printed before program control is returned to the SAP. This printout will not occur if the closed loop response was originally requested.

3. EXAMPLE PROBLEM

The system in Figure 4 is used in this example. The open loop response (Bode stability criteria) printout is shown in Figure 17, and the corresponding plot is shown in Figure 18. For the closed loop response, a gain of 6.5 was used since the root locus plot (Figures 14 and 15) indicated that this gain in the forward loop would give an adequate system response. The printout for hte closed loop response (Bode frequency response) is shown in Figure 19, and the corresponding plot is shown in Figure 20.

4. FLOW CHART

The flow chart for the Bode subroutine is shown in Figure 21. The complete program listing for the subroutine is contained in Appendix E.

```
ROOT LOCUS(L),BODE(B),RESPONSE(R),OR STOP(S)?

BODE PLOT SELECTED

****************
CASE EXAMPLE
****************

OPEN LOOP RESPONSE

NUMERATOR COEFFICIENTS
N( 0     )= 250
N( 1     )= 500
N( 2     )= 250
DENOMINATOR COEFFICIENTS
D( 0     )= 0
D( 1     )= 0
D( 2     )= 0
D( 3     )= 314.159
D( 4     )= 62.8318
D( 5     )= 3.14159

BODE DATA--FREQ(RAD/SEC),FREQ(HZ),LOG AMP(DB),PHASE(DEG)
```

1.000E-01	1.592E-02	5.810E+01	1.003E+02
1.259E-01	2.004E-02	5.215E+01	1.029E+02
1.585E-01	2.522E-02	4.623E+01	1.062E+02
1.995E-01	3.176E-02	4.035E+01	1.103E+02
2.512E-01	3.998E-02	3.454E+01	1.153E+02
3.162E-01	5.033E-02	2.883E+01	1.215E+02
3.981E-01	6.336E-02	2.328E+01	1.288E+02
5.012E-01	7.977E-02	1.794E+01	1.375E+02
6.310E-01	1.004E-01	1.289E+01	1.473E+02
7.943E-01	1.264E-01	8.210E+00	1.578E+02
1.000E+00	1.592E-01	3.950E+00	1.686E+02
1.259E+00	2.004E-01	1.281E-01	1.787E+02
1.585E+00	2.522E-01	-3.289E+00	-1.725E+02
1.995E+00	3.176E-01	-6.377E+00	-1.658E+02
2.512E+00	3.998E-01	-9.238E+00	-1.616E+02
3.162E+00	5.033E-01	-1.198E+01	-1.602E+02
3.981E+00	6.336E-01	-1.473E+01	-1.616E+02
5.012E+00	7.977E-01	-1.759E+01	-1.658E+02
6.310E+00	1.004E+00	-2.068E+01	-1.725E+02
7.943E+00	1.264E+00	-2.410E+01	1.787E+02
1.000E+01	1.592E+00	-2.792E+01	1.686E+02
1.259E+01	2.004E+00	-3.218E+01	1.578E+02
1.585E+01	2.522E+00	-3.686E+01	1.473E+02
1.995E+01	3.176E+00	-4.191E+01	1.375E+02
2.512E+01	3.998E+00	-4.725E+01	1.288E+02
3.162E+01	5.033E+00	-5.280E+01	1.215E+02
3.981E+01	6.336E+00	-5.851E+01	1.153E+02
5.012E+01	7.977E+00	-6.432E+01	1.103E+02
6.310E+01	1.004E+01	-7.020E+01	1.062E+02
7.943E+01	1.264E+01	-7.612E+01	1.029E+02
1.000E+02	1.592E+01	-8.207E+01	1.003E+02

```
AT 0 DB, PHASE= 165.54  DEGREES
AT 180 DEGREES, LOG AMPLITUDE=-0.38    DB
AT 180 DEGREES, LOG AMPLITUDE=-23.59   DB
```

Figure 17. B Subroutine Output for Open Loop-Response

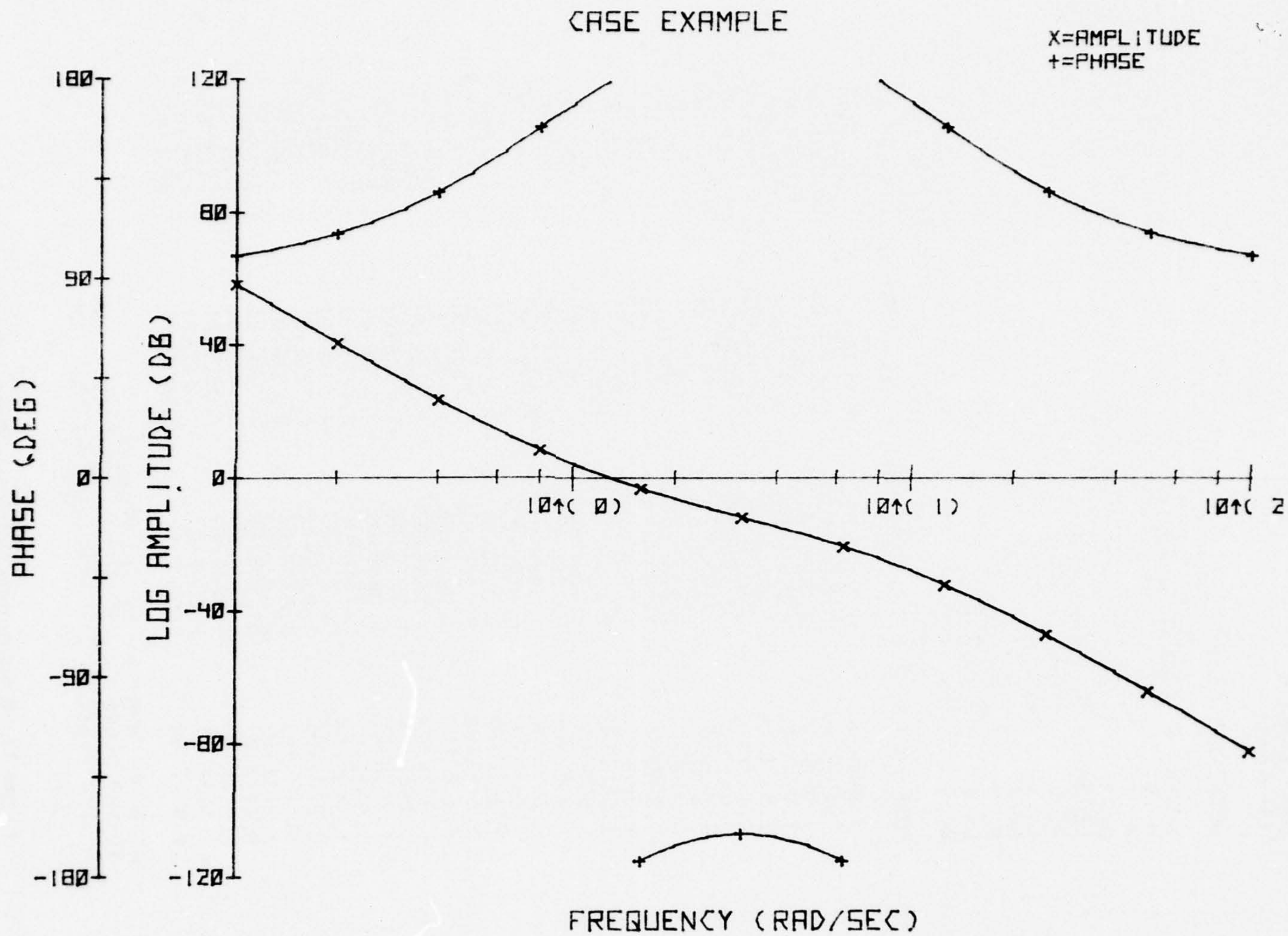

Figure 18. B Subroutine Plot for Open Loop Response

```
ROOT LOCUS(L),BODE(B),RESPONSE(R),OR STOP(S)?

BODE PLOT SELECTED

***************
CASE EXAMPLE
***************

CLOSED LOOP RESPONSE--GAIN= 6.5

NUMERATOR COEFFICIENTS
N( 0     )= 1625
N( 1     )= 1625
DENOMINATOR COEFFICIENTS
D( 0     )= 1625
D( 1     )= 3250
D( 2     )= 1625
D( 3     )= 314.159
D( 4     )= 62.8318
D( 5     )= 3.14159

BODE DATA--FREQ(RAD/SEC),FREQ(HZ),LOG AMP(DB),PHASE(DEG)

 1.000E-01      1.592E-02      -4.292E-02      -5.699E+00
 1.259E-01      2.004E-02      -6.755E-02      -7.153E+00
 1.585E-01      2.522E-02      -1.059E-01      -8.964E+00
 1.995E-01      3.176E-02      -1.651E-01      -1.120E+01
 2.512E-01      3.998E-02      -2.550E-01      -1.395E+01
 3.162E-01      5.033E-02      -3.888E-01      -1.727E+01
 3.981E-01      6.336E-02      -5.816E-01      -2.123E+01
 5.012E-01      7.977E-02      -8.482E-01      -2.585E+01
 6.310E-01      1.004E-01      -1.198E+00      -3.113E+01
 7.943E-01      1.264E-01      -1.632E+00      -3.709E+01
 1.000E+00      1.592E-01      -2.139E+00      -4.377E+01
 1.259E+00      2.004E-01      -2.697E+00      -5.131E+01
 1.585E+00      2.522E-01      -3.273E+00      -5.990E+01
 1.995E+00      3.176E-01      -3.796E+00      -6.988E+01
 2.512E+00      3.998E-01      -4.114E+00      -8.197E+01
 3.162E+00      5.033E-01      -3.904E+00      -9.844E+01
 3.981E+00      6.336E-01      -2.743E+00      -1.282E+02
 5.012E+00      7.977E-01      -4.113E+00       1.668E+02
 6.310E+00      1.004E+00      -1.282E+01       1.175E+02
 7.943E+00      1.264E+00      -2.139E+01       9.502E+01
 1.000E+01      1.592E+00      -2.916E+01       8.030E+01
 1.259E+01      2.004E+00      -3.658E+01       6.832E+01
 1.585E+01      2.522E+00      -4.392E+01       5.775E+01
 1.995E+01      3.176E+00      -5.133E+01       4.827E+01
 2.512E+01      3.998E+00      -5.834E+01       3.985E+01
 3.162E+01      5.033E+00      -6.648E+01       3.255E+01
 3.981E+01      6.336E+00      -7.423E+01       2.636E+01
 5.012E+01      7.977E+00      -8.205E+01       2.121E+01
 6.310E+01      1.004E+01      -8.994E+01       1.699E+01
 7.943E+01      1.264E+01      -9.786E+01       1.357E+01
 1.000E+02      1.592E+01      -1.058E+02       1.082E+01
```

Figure 19. B Subroutine Output for Closed Loop Response

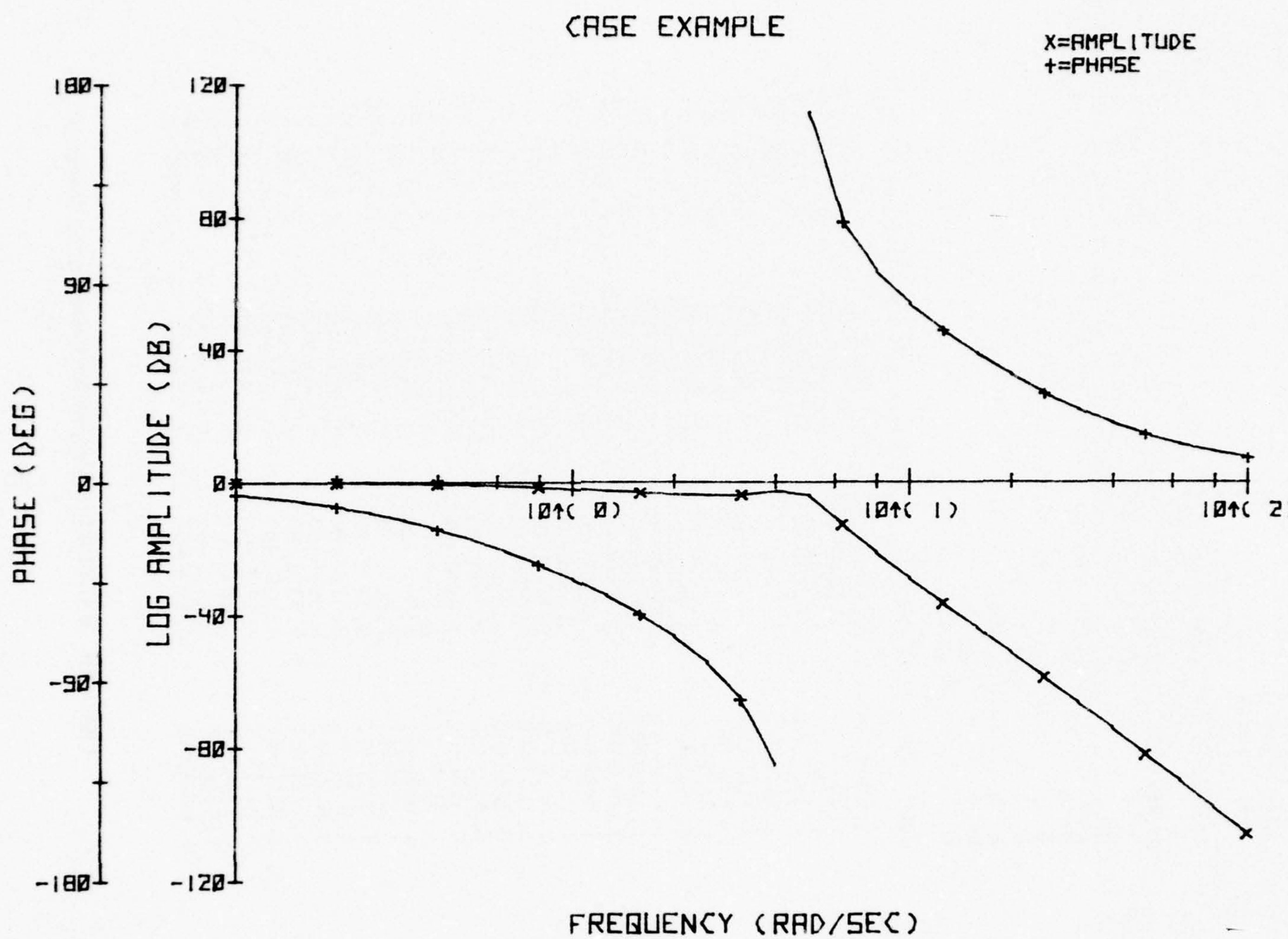

Figure 20. B Subroutine Plot for Closed Loop Response

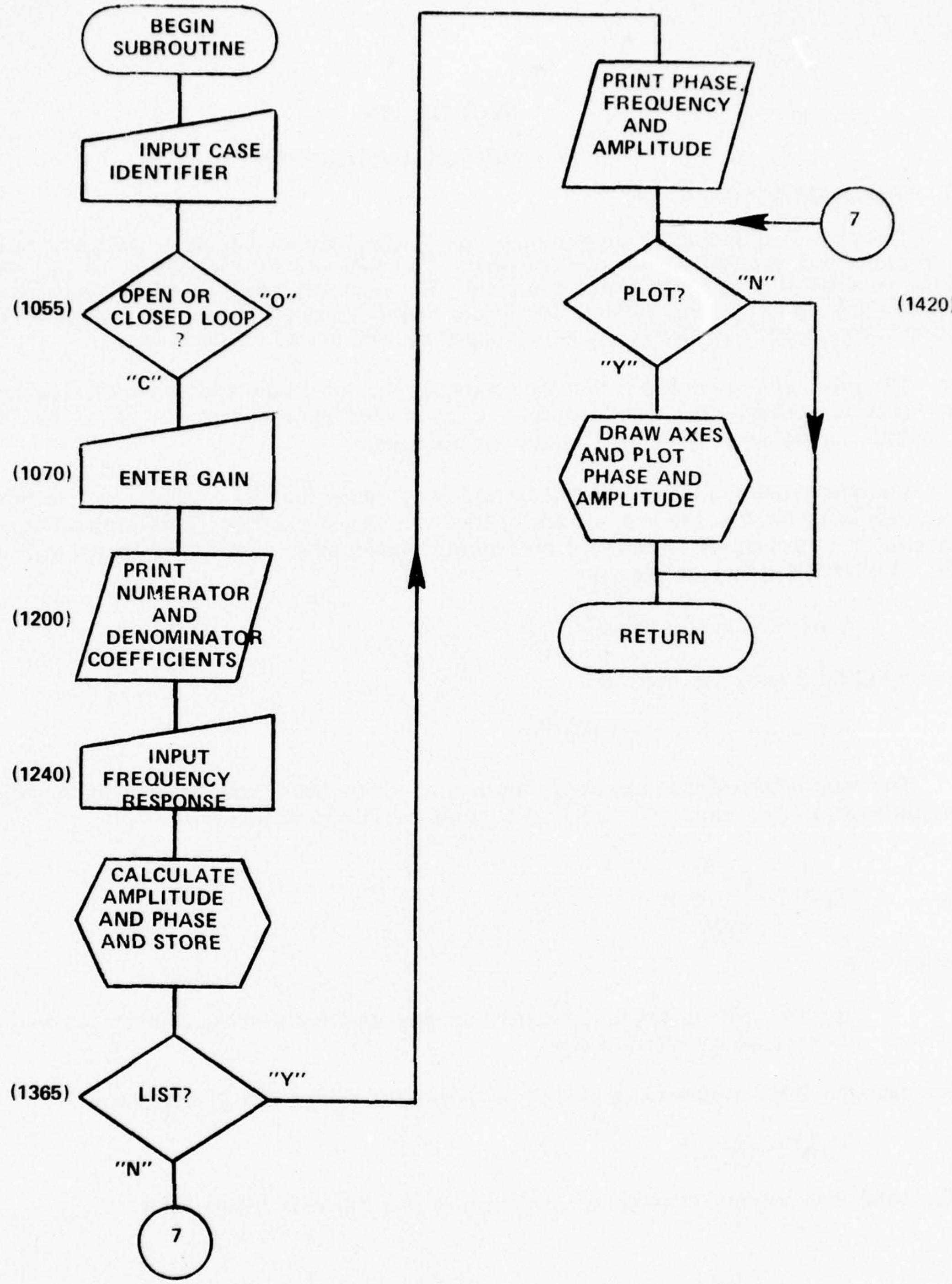

Figure 21. B Subroutine Flow Chart

SECTION VIII

TIME RESPONSE SUBROUTINE

1. PROGRAM DESCRIPTION

The Time Response (TR) Subroutine is used in conjunction with either the SAP-TI executive program or the SAP-KI executive program. This subroutine will calculate and plot the time response of a system to a unit step input. The time response is calculated from inverse Laplace transform. A description of the theory used to calculate the inverse Laplace transform is contained in Appendix G and, consequently, will not be discussed here.

The subroutine is limited in that the subroutine cannot handle a system which has a real root that is repeated more than three times or a repeated complex root pair. Also there is no provision for altering the driving function for the system.

The subroutine prints out information which will allow the user to construct the time response equation, e.g., the inverse Laplace transform, for the system. This information is labeled "Inverse Laplace Transform Information," and an example output is shown in Figure 22. A FORM 1 term is written as

$$y\ (t) = (A + Bt + Ct^2)e^{Rt}$$

and a FORM 2 term is written as

$$y(t) = A \cos It + B \sin It)e^{Rt}$$

The total time response equation is the sum of the particular solution plus the homogeneous terms listed on the output. The particular solution is always of the form

$$Y_p(t) = \frac{b_o\ t^q}{a_q\ q^1}$$

where

q = the order of the lowest order, non-zero coefficient in the denominator of the system transfer function.

For example, if the system has a pole at the origin, i.e., $a_o = o$ and $a_1 \neq o$, then

$$Y_p(t) = \frac{b_o}{a_1}\ t$$

The total time response equation for the example case shown in Figure 23 is

$$t(t) = \frac{N(0)^1}{D(0)} \cdot 0.4185e^{-.5584t} \cdot \left[0.5512 \cos (1.273)t + 0.3172 \sin (1.273)t\right] e$$

$$0.0099 + 0.03805e^{-7.127t} + 0.007745e$$

```
ROOT LOCUS(L),BODE(B),RESPONSE(R),OR STOP(S)?

TIME RESPONSE SELECTED

***************
CASE EXAMPLE
***************

GAIN= 1

NUMERATOR COEFFICIENTS
N( 0     )= 79.57753876
N( 1     )= 79.57753876
DENOMINATOR COEFFICIENTS
D( 0     )= 79.57753876
D( 1     )= 159.1550775
D( 2     )= 79.57753876
D( 3     )= 100
D( 4     )= 20
D( 5     )= 1

*****INVERSE LAPLACE TRANSFORM INFORMATION*****
   FORM          R            I            A            B            C
 1.000E+00   -5.584E-01   0.000E+00   -4.185E-01   0.000E+00   0.000E+00
 2.000E+00    9.946E-03   1.273E+00   -5.512E-01  -3.172E-01   0.000E+00
 1.000E+00   -7.127E+00   0.000E+00   -3.805E-02   0.000E+00   0.000E+00
 1.000E+00   -1.233E+01   0.000E+00    7.745E-03   0.000E+00   0.000E+00

MAX TIME ON PLOT= 6
ABSOLUTE VALUE OF Y MAX= 1.571413843
STEADY STATE VALUE OF Y= 1
```

Figure 22. TR Subroutine Output (Example 1, Gain = 1)

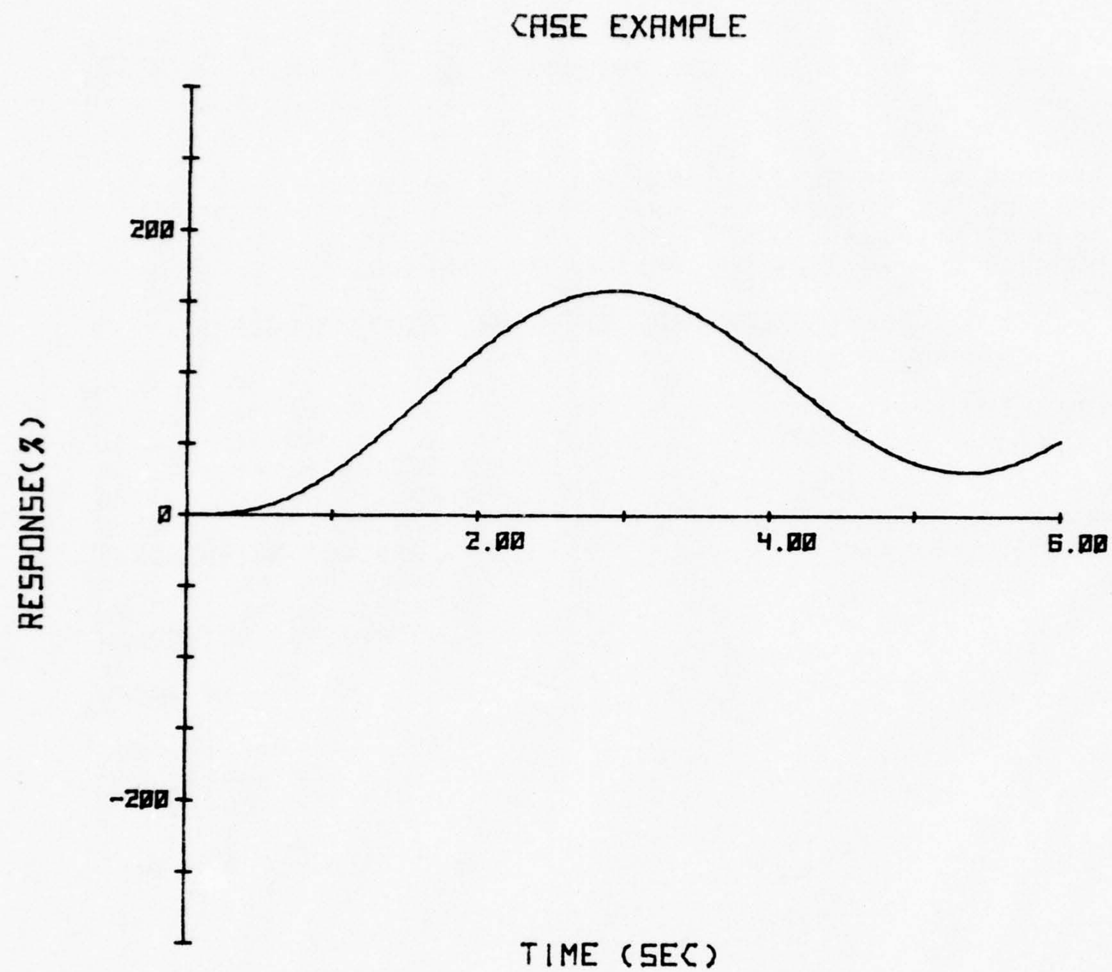

Figure 23. TR Subroutine Plot (Example 1, Gain = 1)

The plot of the time response is automatically scaled and labeled. This is to insure that the plot is readable. To accomplish this scaling, the program will (1) determine the maximum value of the time response of the system in the time interval specified by the user, and (2) determine if the response has some final steady-state value[1]. If the system does have a final steady-state value, the program will scale the plot as a percentage of that final steady-state value, and the y-axis will be labeled RESPONSE (%). If the system does not have a steady-state value, the program will scale the plot based on the maximum value of the response, and the y-axis will be labeled RESPONSE (UNITS). In either case, the scale determined by the program will be labeled on the plot.

2. OPERATING PROCEDURE

a. The display will request a CASE IDENTIFIER. Type in up to ten alphanumeric characters that you wish to use to identify the following plots and printouts. After the identifying characters are entered, press either EXECUTE or STOP to continue program execution. The case identifier will be printed.

b. The display will ask for the gain. This gain will be used as the forward loop gain; so caution must be used when interpreting the results of an isolated loop equivalent (see Section III).

c. The numerator and denominator coefficients of the closed-loop transfer function are printed out. Following this step, the calculator will appear to be inactive for a long period of time while the inverse Laplace transform is calculated.

d. The inverse Laplace transformation will be printed.

e. The display will request the MAX. TIME ON PLOT. The time axis of the plot is six inches long, so if the user wants the tic marks on the plot to coincide with the divisions on graph paper, the maximum time specified should be some multiple of six. After the maximum time is entered, the calculator will appear to be inactive for a long period of time while the system response is calculated for the time interval which was specified.

f. After the maximum time is specified, the user should insure that the plotter is set up.

g. The printer will print out some information, and the response will be plotted.

h. The display will then ask if a new gain is desired, a new time range is desired, or neither of the above. If the response to the question is "N", the control of the program will be returned to the SAP executive program. If a new gain is requested, i.e., "G", the program will return to step b. If a new time scale is specified, i.e., "T", the program will return to step e.

---

[1]The term "steady-state value" means that the term b_0/a_0 is defined. The user should note that when the system is unstable, there is no steady-state value even though the term b_0/a_0 is defined. The program simply bases its decision on the presence of a steady-state value on the existence of a defined b_0/a_0 and not on the stability of the system.

3. EXAMPLE PROBLEMS

a. Example 1

The time response for the system represented by the block diagram in Figure 4 was run for several representative gains. Figures 22 and 23 are the output and plot, respectively, for a forward loop gain of one, which according to the root locus will be a very lightly damped, low frequency oscillation. Figures 24 and 25 are the output and plot for a gain of 6.5. Figures 26 and 27 are the output and plot for a gain of 20, which causes the system to be unstable. The system does have a steady-state value of the response, so the responses are expressed as a percentage of the steady-state value. Note that the scale factor on Figure 25 is different than the scale factor on Figures 23 and 27.

b. Example 2

The time response for the transfer function

$$\frac{S + 1}{S^2 (S + 2) (S + 3)}$$

was run to illustrate the format of the plot for the case where a system has no steady-state value for the time response. Figure 28 is the output of the TR subroutine for this example, and Figure 29 is the plot. The response is expressed in units, and the scale factor is such that each tic mark on the y-axis is one unit. The maximum response occurs at six seconds, and the value of the maximum response is 3.11 units as listed on the output (Figure 28) and shown on the plot (Figure 29).

4. FLOW CHART

The flow chart for the Time Response subroutine is shown in Figure 30. The complete program listing for the subroutine is contained in Appendix F.

```
GAIN= 6.5

NUMERATOR COEFFICIENTS
N( 0      )= 517.2540020
N( 1      )= 517.2540020
DENOMINATOR COEFFICIENTS
D( 0      )= 517.2540020
D( 1      )= 1034.508004
D( 2      )= 517.2540020
D( 3      )= 100
D( 4      )= 20
D( 5      )= 1

*****INVERSE LAPLACE TRANSFORM INFORMATION*****
   FORM          R            I            A            B            C
 1.000E+00   -7.405E-01   0.000E+00   -4.509E-01    0.000E+00    0.000E+00
 1.000E+00   -1.987E+00   0.000E+00   -6.655E-01    0.000E+00    0.000E+00
 2.000E+00   -9.281E-01   4.685E+00    1.058E-01   -2.977E-01    0.000E+00
 1.000E+00   -1.542E+01   0.000E+00    1.058E-02    0.000E+00    0.000E+00

MAX TIME ON PLOT= 6
ABSOLUTE VALUE OF Y MAX= 0.994107991
STEADY STATE VALUE OF Y= 1
```

Figure 24. TR Subroutine Output (Example 1, Gain = 6.5)

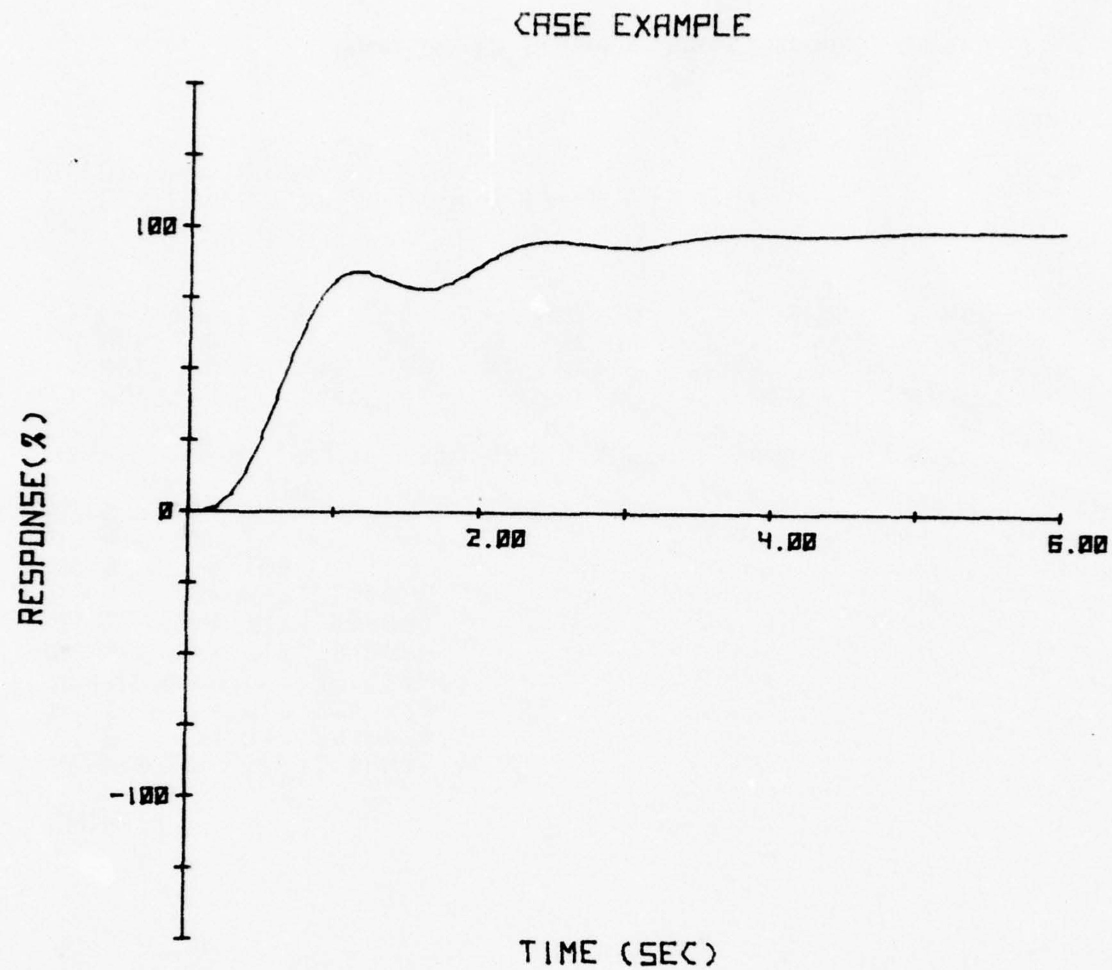

Figure 25. TR Subroutine Plot (Example 1, Gain = 6.5)

```
GAIN= 20

NUMERATOR COEFFICIENTS
N( 0     )= 1591.550775
N( 1     )= 1591.550775
DENOMINATOR COEFFICIENTS
D( 0     )= 1591.550775
D( 1     )= 3183.10155
D( 2     )= 1591.550775
D( 3     )= 100
D( 4     )= 20
D( 5     )= 1

*****INVERSE LAPLACE TRANSFORM INFORMATION*****
   FORM          R              I              A              B              C
 1.000E+00   -8.271E-01     0.000E+00    -4.669E-01     0.000E+00     0.000E+00
 1.000E+00   -1.335E+00     0.000E+00    -5.655E-01     0.000E+00     0.000E+00
 2.000E+00    4.444E-01     8.762E+00     2.156E-02    -1.081E-01     0.000E+00
 1.000E+00   -1.873E+01     0.000E+00     1.089E-02     0.000E+00     0.000E+00

MAX TIME ON PLOT= 6
ABSOLUTE VALUE OF Y MAX= 2.301553578
STEADY STATE VALUE OF Y= 1
```

Figure 26. TR Subroutine Output (Example 1, Gain = 20)

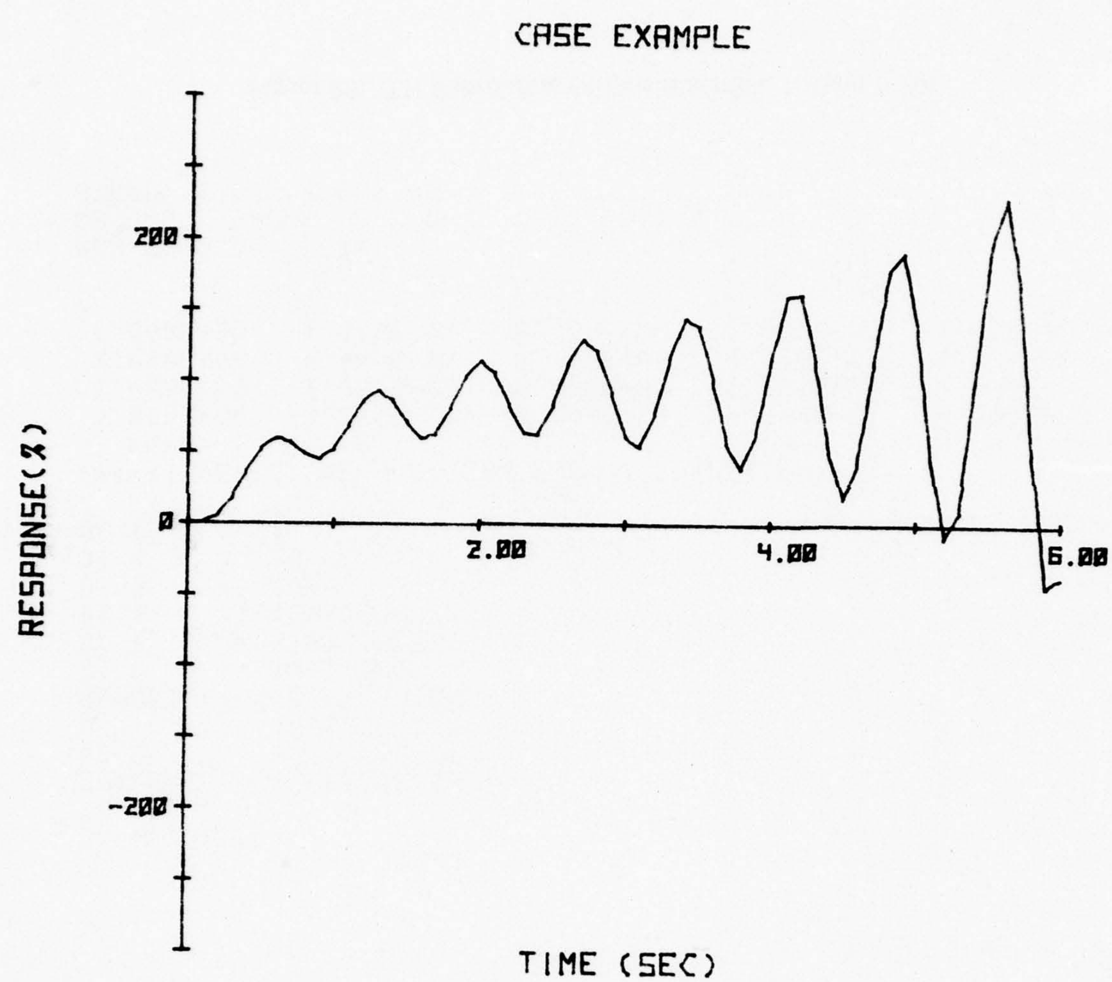

Figure 27. TR Subroutine Plot (Example 1, Gain = 20)

```
**************
CASE FREE POLES
**************

GAIN= 1

NUMERATOR COEFFICIENTS
N( 0     )= 1
N( 1     )= 1
DENOMINATOR COEFFICIENTS
D( 0     )= 0
D( 1     )= 0
D( 2     )= 6
D( 3     )= 5
D( 4     )= 1

*****INVERSE LAPLACE TRANSFORM INFORMATION*****
   FORM          R             I             A            B            C
 1.000E+00   0.000E+00    0.000E+00   -5.093E-02    2.778E-02   0.000E+00
 1.000E+00  -2.000E+00    0.000E+00    1.250E-01    0.000E+00   0.000E+00
 1.000E+00  -3.000E+00    0.000E+00   -7.407E-02    0.000E+00   0.000E+00

MAX TIME ON PLOT= 6
ABSOLUTE VALUE OF Y MAX= 3.115741508
NO STEADY STATE VALUE-- 2    FREE INTEGRATORS
```

Figure 28. TR Subroutine Output (Example 2)

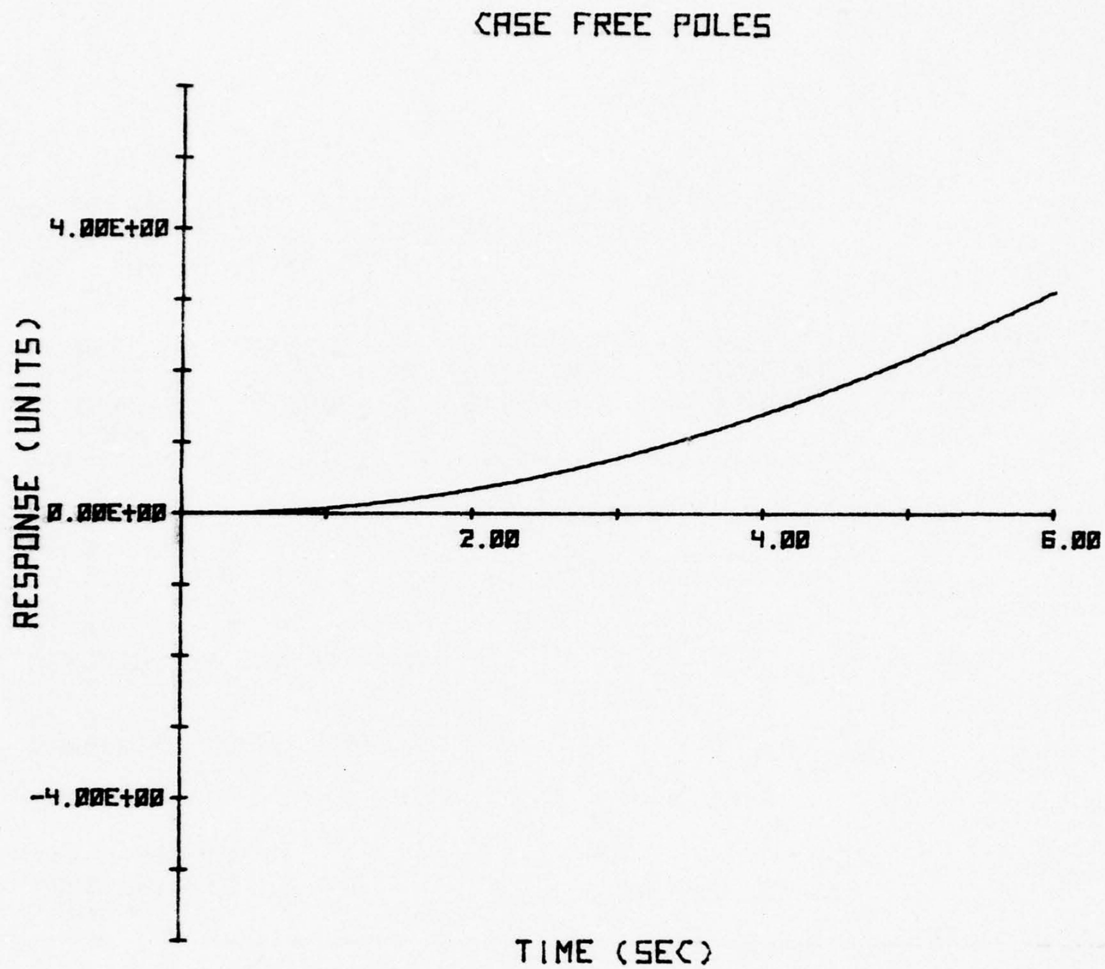

Figure 29. TR Subroutine Plot (Example 2)

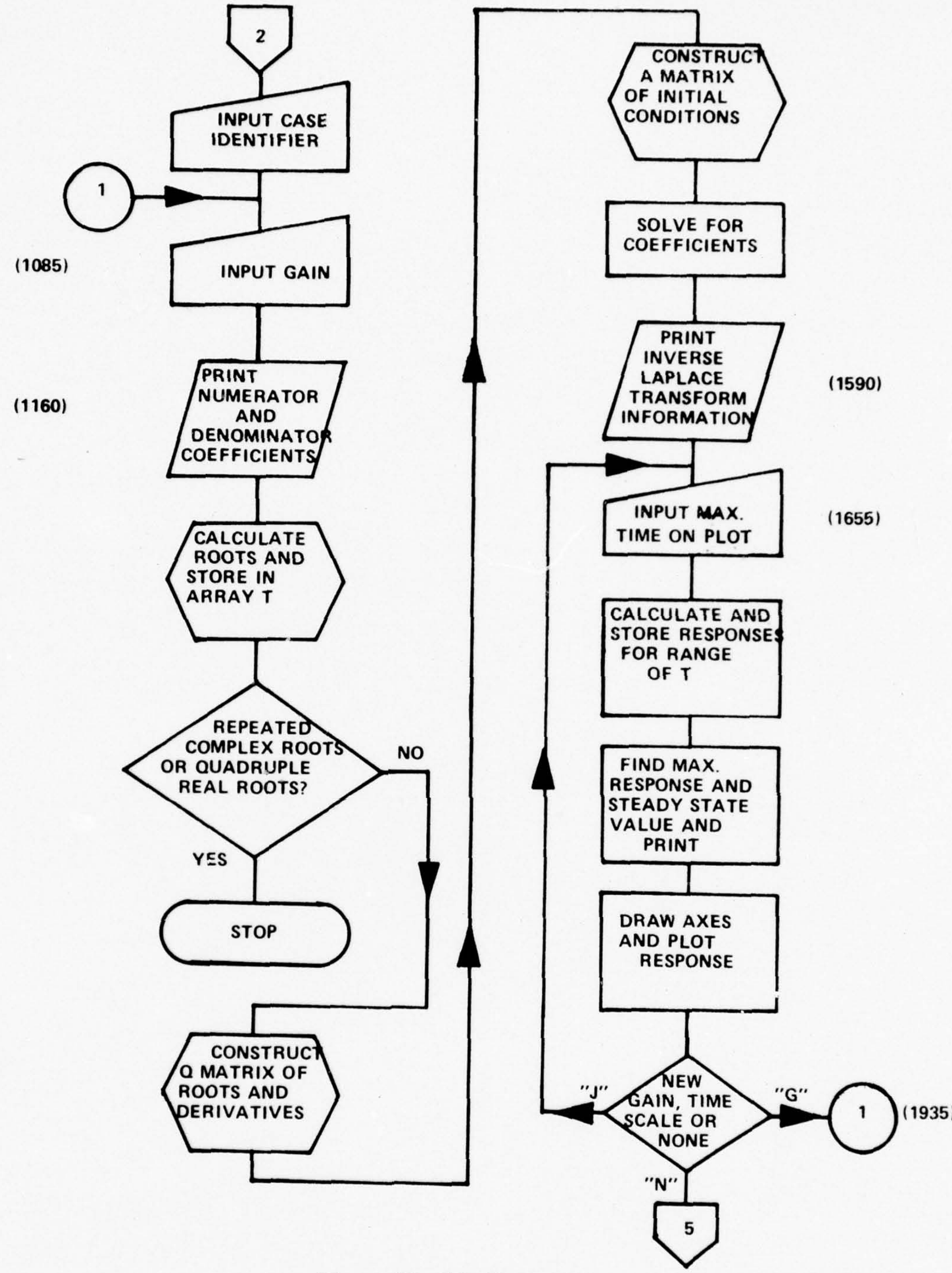

Figure 30. TR Flow Chart

(The reverse of this page is blank)

APPENDIX A

LSP PROGRAM LISTING

```
5 COM U[4,12],V[4,12],W[4,12],X[4,12],Y[4,12],Z[4,12],P[3,12],F6,F7,F8
10 REM*******************
15 REM****LOOP SOLVER****
20 REM*******************
25 DIM P$[1],A[12],B[12],C[12],D[4,12],E[4,12],S[7,12],T[7,12]
30 MAT S=ZER
35 MAT T=ZER
40 MAT U=ZER
45 MAT V=ZER
50 MAT W=ZER
55 MAT X=ZER
60 MAT Y=ZER
65 MAT Z=ZER
70 F6=0
75 F7=0
80 F8=0
85 MAT P=ZER
90 DISP "FORWARD X-FER FUNCTION #";
95 INPUT P
100 IF P=0 THEN 180
105 PRINT "ENTER NUMERATOR DATA FOR G#"P
110 PRINT
115 GOSUB 1595
120 FOR I=1 TO 11
125 S[P,I]=A[I]
130 NEXT I
135 S[P,12]=0
140 PRINT "ENTER DENOMINATOR DATA FOR G#"P
145 PRINT
150 GOSUB 1595
155 FOR I=1 TO 11
160 T[P,I]=A[I]
165 NEXT I
170 T[P,12]=0
175 GOTO 90
180 DISP "FEEDBACK X-FER FUNCTION #";
185 INPUT P
190 IF P=0 THEN 270
```

```
195 PRINT "ENTER NUMERATOR DATA FOR H#"P
200 PRINT
205 GOSUB 1595
210 FOR I=1 TO 11
215 U[P,I]=A[I]
220 NEXT I
225 U[P,12]=0
230 PRINT "ENTER DENOMINATOR DATA FOR H#"P
235 PRINT
240 GOSUB 1595
245 FOR I=1 TO 11
250 V[P,I]=A[I]
255 NEXT I
260 V[P,12]=0
265 GOTO 180
270 PRINT
275 FOR I=1 TO 7
280 IF T[I,1]=0 AND T[I,12]=0 THEN 290
285 GOTO 295
290 T[I,1]=1
295 NEXT I
300 FOR I=1 TO 4
305 IF V[I,1]=0 AND V[I,12]=0 THEN 315
310 GOTO 320
315 V[I,1]=1
320 NEXT I
325 PRINT
330 PRINT "FORWARD LOOP NUMERATORS"
335 FOR I=1 TO 12
340 WRITE (15,345)S[1,I],S[2,I],S[3,I],S[4,I],S[5,I],S[6,I],S[7,I]
345 FORMAT E10.3,X,E10.3,X,E10.3,X,E10.3,X,E10.3,X,E10.3,X,E10.3
350 NEXT I
355 PRINT
360 PRINT "FORWARD LOOP DENOMINATORS"
365 FOR I=1 TO 12
370 WRITE (15,345)T[1,I],T[2,I],T[3,I],T[4,I],T[5,I],T[6,I],T[7,I]
375 NEXT I
380 PRINT
```

```
385 PRINT "FEEDBACK LOOP NUMERATORS"
390 FOR I=1 TO 12
395 WRITE (15,400)U[1,I],U[2,I],U[3,I],U[4,I]
400 FORMAT E10.3,X,E10.3,X,E10.3,X,E10.3
405 NEXT I
410 PRINT
415 PRINT "FEEDBACK LOOP DENOMINATORS"
420 FOR I=1 TO 12
425 WRITE (15,400)V[1,I],V[2,I],V[3,I],V[4,I]
430 NEXT I
435 FOR I=1 TO 12
440 W[1,I]=S[4,I]
445 X[1,I]=T[4,I]
450 D[1,I]=S[4,I]
455 E[1,I]=T[4,I]
460 NEXT I
465 FOR J1=1 TO 3
470 FOR I=1 TO 12
475 D[2,I]=U[J1,I]
480 E[2,I]=V[J1,I]
485 D[3,I]=S[4-J1,I]
490 E[3,I]=T[4-J1,I]
495 D[4,I]=S[4+J1,I]
500 E[4,I]=T[4+J1,I]
505 NEXT I
510 IF D[3,1]=0 AND D[3,12]=0 OR D[4,1]=0 AND D[4,12]=0 THEN 530
515 IF D[1,1]=0 AND D[1,12]=0 THEN 585
520 IF D[2,1]=0 AND D[2,12]=0 THEN 580
525 GOTO 555
530 FOR I=1 TO 12
535 D[1,I]=0
540 E[1,I]=0
545 NEXT I
550 GOTO 585
555 GOSUB 1095
560 FOR I=1 TO 12
565 D[1,I]=A[I]
570 E[1,I]=B[I]+C[I]
```

```
575 NEXT I
580 GOSUB 1225
585 FOR I=1 TO 12
590 W[J1+1,I]=D[1,I]
595 X[J1+1,I]=E[1,I]
600 NEXT I
605 NEXT J1
610 FOR I=1 TO 12
615 Y[4,I]=0
620 Z[4,I]=0
625 D[1,I]=0
630 E[1,I]=0
635 NEXT I
640 Z[4,1]=1
645 E[1,1]=1
650 FOR J1=3 TO 1 STEP -1
655 FOR I=1 TO 12
660 D[2,I]=U[J1+1,I]
665 E[2,I]=V[J1+1,I]
670 D[3,I]=S[4-J1,I]
675 E[3,I]=T[4-J1,I]
680 D[4,I]=S[4+J1,I]
685 E[4,I]=T[4+J1,I]
690 NEXT I
695 IF D[3,1]=0 AND D[3,12]=0 OR D[4,1]=0 AND D[4,12]=0 THEN 705
700 GOTO 730
705 FOR I=1 TO 12
710 D[1,I]=0
715 E[1,I]=0
720 NEXT I
725 GOTO 795
730 IF D[1,1]=0 AND D[1,12]=0 THEN 770
735 IF D[2,1]=0 AND D[2,12]=0 THEN 790
740 GOSUB 1465
745 FOR I=1 TO 12
750 D[1,I]=A[I]+B[I]
755 E[1,I]=C[I]
760 NEXT I
```

```
765 GOTO 790
770 FOR I=1 TO 12
775 E[1,I]=E[2,I]
780 D[1,I]=D[2,I]
785 NEXT I
790 GOSUB 1225
795 FOR I=1 TO 12
800 Y[J1,I]=D[1,I]
805 Z[J1,I]=E[1,I]
810 NEXT I
815 NEXT J1
820 FOR J1=1 TO 4
825 FOR I=1 TO 12
830 D[1,I]=W[J1,I]
835 E[1,I]=X[J1,I]
840 D[2,I]=Y[J1,I]
845 E[2,I]=Z[J1,I]
850 NEXT I
855 IF D[1,1]=0 AND D[1,12]=0 OR D[2,1]=0 AND D[2,12]=0 THEN 890
860 GOSUB 1095
865 FOR I=1 TO 12
870 Y[J1,I]=A[I]
875 Z[J1,I]=B[I]+C[I]
880 NEXT I
885 GOTO 910
890 FOR I=1 TO 12
895 Y[J1,I]=D[1,I]
900 Z[J1,I]=E[1,I]
905 NEXT I
910 NEXT J1
915 PRINT
920 PRINT
925 PRINT "INTERNAL LOOP EQUIVALENT FORWARD NUMERATORS"
930 FOR I=1 TO 12
935 WRITE (15,400)W[1,I],W[2,I],W[3,I],W[4,I]
940 NEXT I
945 FOR I=1 TO 4
950 IF X[I,1]=0 AND X[I,12]=0 THEN 960
```

```
955 GOTO 965
960 X[I,1]=1
965 NEXT I
970 PRINT
975 PRINT "INTERNAL LOOP EQUIVALENT FORWARD DENOMINATORS"
980 FOR I=1 TO 12
985 WRITE (15,400)X[1,I],X[2,I],X[3,I],X[4,I]
990 NEXT I
995 PRINT
1000 PRINT "ISOLATED LOOP EQUIVALENT FORWARD NUMERATORS"
1005 FOR I=1 TO 12
1010 WRITE (15,400)Y[1,I],Y[2,I],Y[3,I],Y[4,I]
1015 NEXT I
1020 FOR I=1 TO 4
1025 IF Z[I,1]=0 AND Z[I,12]=0 THEN 1035
1030 GOTO 1040
1035 Z[I,1]=1
1040 NEXT I
1045 PRINT
1050 PRINT "ISOLATED LOOP EQUIVALENT FORWARD DENOMINATORS"
1055 FOR I=1 TO 12
1060 WRITE (15,400)Z[1,I],Z[2,I],Z[3,I],Z[4,I]
1065 NEXT I
1070 DISP "DATA STORAGE FILE NO.=";
1075 INPUT P1
1080 STORE  DATA P1
1085 GOTO 50
1090 END
1095 MAT A=ZER
1100 MAT B=ZER
1105 MAT C=ZER
1110 O=D[1,12]+E[2,12]
1115 A[12]=O
1120 O1=E[1,12]+E[2,12]
1125 B[12]=O1
1130 IF O>O1 THEN 1140
1135 O=O1
1140 O1=D[1,12]+D[2,12]
1145 C[12]=O1
1150 IF O>O1 THEN 1160
1155 O=O1
1160 FOR I=1 TO O+1
1165 FOR J=1 TO I
1170 I2=I-J+1
1175 A[I]=A[I]+D[1,J]*E[2,I2]
1180 B[I]=B[I]+E[1,J]*E[2,I2]
1185 C[I]=C[I]+D[1,J]*D[2,I2]
1190 NEXT J
1195 NEXT I
1200 IF B[12]>C[12] THEN 1210
```

```
1205 B[12]=C[12]
1210 C[12]=0
1215 RETURN
1220 END
1225 MAT A=ZER
1230 MAT B=ZER
1235 MAT C=ZER
1240 FOR I=1 TO 12
1245 A[I]=D[1,I]
1250 B[I]=D[3,I]
1255 NEXT I
1260 O=A[12]+B[12]
1265 GOSUB 1425
1270 MAT A=C
1275 MAT C=ZER
1280 FOR I=1 TO 12
1285 B[I]=D[4,I]
1290 NEXT I
1295 O=O+D[4,12]
1300 GOSUB 1425
1305 FOR I=1 TO 11
1310 D[1,I]=C[I]
1315 NEXT I
1320 D[1,12]=0
1325 MAT C=ZER
1330 FOR I=1 TO 12
1335 A[I]=E[1,I]
1340 B[I]=E[3,I]
1345 NEXT I
1350 O=A[12]+B[12]
1355 GOSUB 1425
1360 MAT A=C
1365 MAT C=ZER
1370 FOR I=1 TO 12
1375 B[I]=E[4,I]
1380 NEXT I
1385 O=O+E[4,12]
1390 GOSUB 1425
1395 FOR I=1 TO 11
1400 E[1,I]=C[I]
1405 NEXT I
1410 E[1,12]=0
1415 RETURN
1420 END
1425 FOR I=1 TO O+1
1430 FOR J=1 TO I
1435 I2=I-J+1
1440 C[I]=C[I]+A[J]*B[I2]
1445 NEXT J
1450 NEXT I
```

```
1455 RETURN
1460 END
1465 MAT A=ZER
1470 MAT B=ZER
1475 MAT C=ZER
1480 O=D[1,12]+E[2,12]
1485 A[12]=O
1490 O1=D[2,12]+E[1,12]
1495 B[12]=O1
1500 IF O>O1 THEN 1510
1505 O=O1
1510 O1=E[1,12]+E[2,12]
1515 C[12]=O1
1520 IF O>O1 THEN 1530
1525 O=O1
1530 FOR I=1 TO O+1
1535 FOR J=1 TO I
1540 I2=I-J+1
1545 A[I]=A[I]+D[1,J]*E[2,I2]
1550 B[I]=B[I]+D[2,J]*E[1,I2]
1555 C[I]=C[I]+E[1,J]*E[2,I2]
1560 NEXT J
1565 NEXT I
1570 IF A[12]>B[12] THEN 1580
1575 A[12]=B[12]
1580 B[12]=0
1585 RETURN
1590 END
1595 REM*********************************************
1600 REM****POLYNOMINAL INPUT & MULTIPLICATION SUBROUTINE**
1605 REM*********************************************
1610 MAT C=ZER
1615 MAT A=ZER
1620 MAT B=ZER
1625 F1=0
1630 DISP "ORDER OF POLYNOMIAL=";
1635 INPUT O
1640 IF O=0 THEN 1760
1645 FOR I=1 TO O+1
1650 DISP "A("I-1")=";
1655 INPUT A[I]
1660 NEXT I
1665 F1=1
1670 DISP "ORDER OF POLYNOMIAL=";
1675 INPUT D1
1680 IF D1=0 THEN 1760
1685 O=O+D1
1690 FOR I=1 TO D1+1
1695 DISP "A("I-1")=";
1700 INPUT B[I]
```

```
1705 NEXT I
1710 FOR I=1 TO O-1
1715 FOR J=1 TO I
1720 I2=I-J+1
1725 C[I]=C[I]+A[J]*B[I2]
1730 NEXT J
1735 NEXT I
1740 MAT A=C
1745 MAT B=ZER
1750 MAT C=ZER
1755 GOTO 1670
1760 DISP "CONSTANT=";
1765 INPUT D9
1770 IF D9=0 THEN 1805
1775 IF F1>0 THEN 1795
1780 F1=1
1785 A[1]=D9
1790 GOTO 1670
1795 MAT A=(D9)*A
1800 GOTO 1670
1805 RETURN
1810 END
```

APPENDIX B

SAP-TI PROGRAM LISTING

```
5 COM U[4,12],V[4,12],W[4,12],X[4,12],Y[4,12],Z[4,12],P[3,12],F6,F7,F8
10 REM**********************************************************
15 REM****STABILITY ANALYSIS EXECUTIVE PROGRAM--TAPE INPUT****
20 REM**********************************************************
25 DIM I$[1],P$[1],D$[1],O[10,20],A[10],C[10],S[7,12],T[7,12],T$[10]
30 F6=0
35 F4=0
40 DISP "DATA FILE NO.";
45 INPUT P1
50 LOAD  DATA P1
55 F6=0
60 DISP "INTERNAL(1) OR ISOLATED(2) LOOP";
65 INPUT P1
70 DISP "NODE NO.";
75 INPUT L
80 GOSUB 245
85 PRINT
90 PRINT "ROOT LOCUS(L),BODE(B),RESPONSE(R),OR STOP(S)?"
95 PRINT
100 INPUT I$
105 IF I$#"L" THEN 125
110 PRINT "ROOT LOCUS SELECTED"
115 F8=3
120 GOTO 175
125 IF I$#"B" THEN 145
130 PRINT "BODE PLOT SELECTED"
135 F8=4
140 GOTO 175
145 IF I$#"R" THEN 165
150 PRINT "TIME RESPONSE SELECTED"
155 F8=5
160 GOTO 175
165 IF I$#"S" THEN 90
170 STOP
175 PRINT
180 IF F8=F6 OR F8=F4 THEN 190
185 LOAD F8,1000
190 F4=F8
```

```
195 F6=F5
200 GOSUB 1000
205 DISP "NEW TRANSFER FUNCTION--Y OR N";
210 INPUT P$
215 IF P$="N" THEN 90
220 DISP "NEW(N) OR EXISTING(E) FILE";
225 INPUT D$
230 IF D$="N" THEN 40
235 GOTO 60
240 END
245 REM**************************
250 REM*****INPUT SUBROUTINE*****
255 REM**************************
260 IF P1=2 THEN 290
265 FOR I=1 TO 12
270 S[4,I]=W[L,I]
275 S[5,I]=X[L,I]
280 NEXT I
285 GOTO 310
290 FOR I=1 TO 12
295 S[4,I]=Y[L,I]
300 S[5,I]=Z[L,I]
305 NEXT I
310 FOR I=1 TO 12
315 S[6,I]=U[L,I]
320 S[7,I]=V[L,I]
325 NEXT I
330 MAT P=ZER
335 P[1,12]=S[4,12]+S[7,12]
340 P[2,12]=S[4,12]+S[6,12]
345 P[3,12]=S[5,12]+S[7,12]
350 IF P[3,12]>P[1,12] AND P[2,12]>P[3,12] THEN 370
355 IF P[3,12]>P[1,12] AND P[3,12]>P[2,12] THEN 380
360 O=P[1,12]
365 GOTO 385
370 O=P[2,12]
375 GOTO 385
380 O=P[3,12]
385 FOR I=1 TO O+1
390 FOR J=1 TO I
395 I2=I-J+1
400 P[1,I]=P[1,I]+S[4,J]*S[7,I2]
405 P[2,I]=P[2,I]+S[4,J]*S[6,I2]
410 P[3,I]=P[3,I]+S[5,J]*S[7,I2]
415 NEXT J
420 NEXT I
425 RETURN
430 END
```

APPENDIX C

SAP-KI PROGRAM LISTING

```
5 COM F[3,12],F6,F7,F8
10 REM*************************************************************
15 REM****STABILITY ANALYSIS EXECUTIVE PROGRAM--KEYBOARD INPUT****
20 REM*************************************************************
25 DIM I$[1],P$[1],D$[1],Q[10,30],A[10],C[10],S[7,12],T[7,12],T$[10
30 F6=0
35 GOSUB 180
40 PRINT
45 PRINT "ROOT LOCUS(L),BODE(B),RESPONSE(R),OR STOP(S)?"
50 PRINT
55 INPUT I$
60 IF I$#"L" THEN 80
65 PRINT "ROOT LOCUS SELECTED"
70 F8=3
75 GOTO 130
80 IF I$#"B" THEN 100
85 PRINT "BODE PLOT SELECTED"
90 F8=4
95 GOTO 130
100 IF I$#"R" THEN 120
105 PRINT "TIME RESPONSE SELECTED"
110 F8=5
115 GOTO 130
120 IF I$#"S" THEN 45
125 STOP
130 PRINT
135 IF F8=F6 THEN 150
140 F6=F8
145 LOAD F6,1000
150 GOSUB 1000
155 DISP "NEW TRANSFER FUNCTION--Y OR N";
160 INPUT P$
165 IF P$="N" THEN 45
170 GOTO 35
175 END
180 REM**********************************
185 REM****KEYBOARD INPUT SUBROUTINE****
190 REM**********************************
```

```
195 MAT S=ZER
200 MAT T=ZER
205 PRINT "ENTER NUMERATOR OF G"
210 PRINT
215 GOSUB 430
220 FOR I=1 TO 12
225 S[4,I]=S[3,I]
230 NEXT I
235 PRINT "ENTER DENOMINATOR OF G"
240 PRINT
245 GOSUB 430
250 FOR I=1 TO 12
255 S[5,I]=S[3,I]
260 NEXT I
265 PRINT "ENTER NUMERATOR OF H"
270 PRINT
275 GOSUB 430
280 FOR I=1 TO 12
285 S[6,I]=S[3,I]
290 NEXT I
295 PRINT "ENTER DENOMINATOR OF H"
300 PRINT
305 GOSUB 430
310 FOR I=1 TO 12
315 S[7,I]=S[3,I]
320 NEXT I
325 MAT P=ZER
330 P[1,12]=S[4,12]+S[7,12]
335 P[2,12]=S[4,12]+S[6,12]
340 P[3,12]=S[5,12]+S[7,12]
345 IF P[2,12]>P[1,12] AND P[2,12]>P[3,12] THEN 365
350 IF P[3,12]>P[1,12] AND P[3,12]>P[2,12] THEN 375
355 O=P[1,12]
360 GOTO 380
365 O=P[2,12]
370 GOTO 380
375 O=P[3,12]
380 FOR I=1 TO O+1
385 FOR J=1 TO I
390 I2=I-J+1
395 P[1,I]=P[1,I]+S[4,J]*S[7,I2]
400 P[2,I]=P[2,I]+S[4,J]*S[6,I2]
405 P[3,I]=P[3,I]+S[5,J]*S[7,I2]
410 NEXT J
415 NEXT I
420 RETURN
425 END
430 REM**************************************************************
435 REM****POLYNOMINAL INPUT & MULTIPLICATION SUBROUTINE****
440 REM**************************************************************
```

```
445 FOR I=1 TO 12
450 S[3,I]=0
455 S[1,I]=0
460 S[2,I]=0
465 NEXT I
470 E=0
475 DISP "ORDER OF POLYNOMIAL=";
480 INPUT O
485 IF O=0 THEN 615
490 FOR I=1 TO O+1
495 DISP "A("I-1")=";
500 INPUT S[3,I]
505 NEXT I
510 E=1
515 DISP "ORDER OF POLYNOMIAL=";
520 INPUT D1
525 IF D1=0 THEN 615
530 O=O+D1
535 FOR I=1 TO D1+1
540 DISP "A("I-1")=";
545 INPUT S[1,I]
550 NEXT I
555 FOR I=1 TO O+1
560 FOR J=1 TO I
565 I2=I-J+1
570 S[2,I]=S[2,I]+S[3,J]*S[1,I2]
575 NEXT J
580 NEXT I
585 FOR I=1 TO 12
590 S[3,I]=S[2,I]
595 S[1,I]=0
600 S[2,I]=0
605 NEXT I
610 GOTO 515
615 DISP "CONSTANT=";
620 INPUT D9
625 IF D9=0 THEN 670
630 IF E>0 THEN 650
635 E=1
640 S[3,1]=D9
645 GOTO 515
650 FOR I=1 TO 12
655 S[3,I]=D9*S[3,I]
660 NEXT I
665 GOTO 515
670 S[3,12]=0
675 RETURN
680 END
```

APPENDIX D

RL SUBROUTINE LISTING

```
1005 REM****ROOT LOCUS EXECUTIVE SUBROUTINE*****
1010 REM**************************************
1015 REDIM Q[3,66]
1020 DISP "ENTER CASE IDENTIFIER";
1025 INPUT T$
1030 PRINT "***************"
1035 PRINT "CASE "T$
1040 PRINT "***************"
1045 MAT S=ZER
1050 MAT Q=ZER
1055 D$="N"
1060 STANDARD
1065 PRINT "NUMERATOR COEFFICIENTS"
1070 FOR I=1 TO P[2,12]+1
1075 PRINT "N("I-1")="P[2,I]
1080 NEXT I
1085 PRINT "DENOMINATOR COEFFICIENTS"
1090 FOR I=1 TO P[3,12]+1
1095 PRINT "D("I-1")="P[3,I]
1100 NEXT I
1105 FIXED 2
1110 DISP "IS PLOT DESIRED--Y OR N";
1115 INPUT P$
1120 IF P$="N" THEN 1160
1125 I0=1
1130 GOSUB 1905
1135 IF D$="N" THEN 1160
1140 FOR I0=1 TO I9
1145 GOSUB 2145
1150 NEXT I0
1155 GOTO 1365
1160 L=1
1165 PRINT
1170 PRINT "ZEROS"
1175 FOR I=1 TO 11
1180 S[3,I]=P[2,I]
1185 NEXT I
1190 O=P[2,12]
1195 IF O#0 THEN 1210
1200 PRINT "    NONE"
1205 GOTO 1215
1210 GOSUB 1390
1215 PRINT
1220 PRINT "POLES"
1225 L=2
1230 FOR I=1 TO 11
1235 S[3,I]=P[3,I]
1240 NEXT I
1245 O=P[3,12]
```

```
1250 GOSUB 1390
1255 IF P[2,12]>P[3,12] THEN 1270
1260 D0=P[3,12]
1265 GOTO 1275
1270 D0=P[2,12]
1275 DISP "GAIN-MIN,MAX,INCR=";
1280 INPUT Z0,Z1,Z2
1285 L=3
1290 PRINT
1295 FOR Z=Z0 TO Z1 STEP Z2
1300 PRINT
1305 PRINT "GAIN="Z
1310 FOR I=1 TO 11
1315 S[3,I]=Z*P[2,I]+P[3,I]
1320 NEXT I
1325 O=D0
1330 GOSUB 1390
1335 NEXT Z
1340 DISP "NEW GAIN INCREMENT--Y OR N";
1345 INPUT I$
1350 IF I$="Y" THEN 1275
1355 IF P$="N" THEN 1380
1360 I9=I0
1365 DISP "A REPLOT--Y OR N";
1370 INPUT D$
1375 IF D$="Y" THEN 1130
1380 RETURN
1385 END
1390 REM**********************************
1395 REM****POLYNOMIAL ROOT SUBROUTINE****
1400 REM**********************************
1405 FOR I=1 TO O+1
1410 S[3,I]=S[3,I]/S[3,O+1]
1415 NEXT I
1420 R1=0
1425 R=0
1430 S=0
1435 IF O<2 THEN 1875
1440 IF S[3,1]#0 THEN 1490
1445 X=0
1450 Y=0
1455 T=1
1460 GOSUB 2145
1465 O=O-1
1470 FOR I=1 TO O+2
1475 S[3,I]=S[3,I+1]
1480 NEXT I
1485 GOTO 1435
1490 IF S#0 THEN 1505
1495 R=0
```

```
1500 GOTO 1515
1505 R=R/S
1510 S=1/S
1515 IF R1=0 THEN 1525
1520 R1=1/R1
1525 S1=5*10↑(-10)
1530 S[4,O+1]=S[3,O+1]
1535 S[5,O+1]=S[3,O+1]
1540 S[4,O+2]=0
1545 S[5,O+2]=0
1550 FOR I2=1 TO 20
1555 FOR I1=0 TO 1 STEP -1
1560 S[4,I1]=S[3,I1]+R1*S[4,I1+1]
1565 S[5,I1]=S[4,I1]+R1*S[5,I1+1]
1570 NEXT I1
1575 IF (ABS(S[4,1]/S[3,1])-S1)>0 THEN 1625
1580 Y=0
1585 X=R1
1590 T=1
1595 GOSUB 2145
1600 O=O-1
1605 FOR I=1 TO O+1
1610 S[3,I]=S[4,I+1]
1615 NEXT I
1620 GOTO 1490
1625 IF S[5,2]#0 THEN 1640
1630 R1=R1+1
1635 GOTO 1645
1640 R1=R1-S[4,1]/S[5,2]
1645 FOR I=0 TO 1 STEP -1
1650 S[4,I]=S[3,I]-R*S[4,I+1]-S*S[4,I+2]
1655 S[5,I]=S[4,I]-R*S[5,I+1]-S*S[5,I+2]
1660 NEXT I
1665 IF S[3,2]#0 THEN 1680
1670 IF (ABS(S[4,2]/S[3,1])-S1)>0 THEN 1690
1675 GOTO 1685
1680 IF (ABS(S[4,2]/S[3,2])-S1)>0 THEN 1690
1685 IF (ABS(S[4,1]/S[3,1])-S1) <= 0 THEN 1755
1690 T=S[5,2]-S[4,2]
1695 U=S[5,3]↑2-T*S[5,4]
1700 IF U#0 THEN 1720
1705 R=R-2
1710 S=S*(S+1)
1715 GOTO 1730
1720 R=R+(S[4,2]*S[5,3]-S[4,1]*S[5,4])/U
1725 S=S+(-S[4,2]*T+S[4,1]*S[5,3])/U
1730 NEXT I2
1735 S1=S1*10
1740 IF S1#5*10↑(-6) THEN 1550
1745 PRINT "TOLERANCE=5*10(-6),NO ROOTS"
```

```
1750 STOP
1755 G=R↑2-4*S
1760 IF G<0 THEN 1805
1765 X=-R/2+SQR(G)/2
1770 Y=0
1775 T=1
1780 GOSUB 2145
1785 X=-R/2-SQR(G)/2
1790 T=1
1795 GOSUB 2145
1800 GOTO 1825
1805 X=-R/2
1810 Y=SQR(-G)/2
1815 T=2
1820 GOSUB 2145
1825 O=O-2
1830 IF O=0 THEN 1895
1835 FOR I=O+1 TO 1 STEP -1
1840 S[3,I]=S[4,I+2]
1845 NEXT I
1850 IF O>2 THEN 1490
1855 IF O<2 THEN 1875
1860 R=S[3,O]/S[3,O+1]
1865 S=S[3,O-1]/S[3,O+1]
1870 GOTO 1755
1875 X=-S[3,O]/S[3,O+1]
1880 Y=0
1885 T=1
1890 GOSUB 2145
1895 RETURN
1900 END
1905 REM********************************************
1910 REM****ROOT LOCUS AXES & SCALING SUBROUTINE****
1915 REM********************************************
1920 SCALE -80,20,-10,60
1925 DISP "MAX REAL VALUE=";
1930 INPUT S0
1935 S0=ABS(S0)
1940 DISP "MAX IMAGINARY VALUE=";
1945 INPUT J0
1950 J0=ABS(J0)
1955 DISP "REAL INCREMENT=";
1960 INPUT S1
1965 S1=ABS(S1)
1970 DISP "IMAGINARY INCREMENT=";
1975 INPUT J1
1980 J1=ABS(J1)
1985 XAXIS 0,60*S1/S0,-60,0
1990 YAXIS 0,50*J1/J0,0,50
1995 LABEL (*,1.25,1.7,0,7/10)
```

```
2000 FOR X=S0 TO 0 STEP -S1
2005 IF X=0 THEN 2025
2010 PLOT -60*X/S0,0,1
2015 CPLOT -2.8,-1.3
2020 LABEL (2060)-X
2025 NEXT X
2030 FOR Y=0 TO J0 STEP J1
2035 IF Y=0 THEN 2055
2040 PLOT 0,50*Y/J0,1
2045 CPLOT 0.3,-0.3
2050 LABEL (2060)Y
2055 NEXT Y
2060 FORMAT 1F6.2
2065 LABEL (*,2,1.7,0,7/10)
2070 PLOT -30,0,1
2075 CPLOT -2,-3
2080 LABEL (*)"REAL"
2085 PLOT 0,25,1
2090 LABEL (*,2,1.7,PI/2,7/10)
2095 CPLOT -4.5,-4
2100 LABEL (*)"IMAGINARY"
2105 LABEL (*,2,1.7,0,7/10)
2110 PLOT -30,60,1
2115 B=LEN(T$)
2120 A=-((B+5)/2)+0.2
2125 CPLOT A,-0.6
2130 LABEL (*)"CASE "T$
2135 RETURN
2140 END
2145 REM*************************************************
2150 REM****CHARACTER PLOTTING SUBROUTINE****
2155 REM*************************************************
2160 IF D$="Y" THEN 2235
2165 PRINT X,Y
2170 IF T=1 THEN 2180
2175 PRINT X,-Y
2180 IF P$="N" THEN 2340
2185 IF X>0.1*S0 THEN 2260
2190 IF Y<0 THEN 2280
2195 IF I0 >= 66 THEN 2225
2200 Q[3,I0]=L
2205 Q[1,I0]=X
2210 Q[2,I0]=Y
2215 I0=I0+1
2220 GOTO 2250
2225 I0=66
2230 GOTO 2250
2235 L=Q[3,I0]
2240 X=Q[1,I0]
2245 Y=Q[2,I0]
```

```
2250 IF X<-S0 THEN 2280
2255 IF Y>J0 THEN 2280
2260 PLOT 60*X/S0,50+Y/J0,1
2265 IF L=1 THEN 2285
2270 IF L=2 THEN 2305
2275 IF L=3 THEN 2325
2280 RETURN
2285 LABEL (*,1.5,1,0,7/10)
2290 CPLOT -0.3,-0.3
2295 LABEL (*)"O"
2300 RETURN
2305 LABEL (*,1.5,1,0,7/10)
2310 CPLOT -0.3,-0.3
2315 LABEL (*)"X"
2320 RETURN
2325 LABEL (*,1,1,0,7/10)
2330 CPLOT -0.3,-0.3
2335 LABEL (*)"X"
2340 RETURN
2345 END
```

APPENDIX E

B SUBROUTINE LISTING

```
1000 REM****************************************
1005 REM*****BODE PLOT EXECUTIVE SUBROUTINE*****
1010 REM****************************************
1015 REDIM Q[3,31]
1020 DISP "ENTER CASE IDENTIFIER";
1025 INPUT T$
1030 PRINT "***************"
1035 PRINT "CASE "T$
1040 PRINT "***************"
1045 MAT S=ZER
1050 STANDARD
1055 DISP "OPEN(O) OR CLOSED(C) LOOP";
1060 INPUT D$
1065 IF D$="O" THEN 1145
1070 DISP "ENTER GAIN";
1075 INPUT K
1080 PRINT
1085 PRINT "CLOSED LOOP RESPONSE--GAIN="K
1090 PRINT
1095 FOR I=1 TO 11
1100 S[1,I]=K*P[1,I]
1105 S[2,I]=K*P[2,I]+P[3,I]
1110 NEXT I
1115 S[1,12]=P[1,12]
1120 IF P[3,12]>P[2,12] THEN 1135
1125 S[2,12]=P[2,12]
1130 GOTO 1140
1135 S[2,12]=P[3,12]
1140 GOTO 1180
1145 PRINT
1150 PRINT "OPEN LOOP RESPONSE"
1155 PRINT
1160 FOR I=1 TO 12
1165 S[1,I]=P[2,I]
1170 S[2,I]=P[3,I]
1175 NEXT I
1180 IF S[1,12]<S[2,12] THEN 1195
1185 O=S[1,12]
1190 GOTO 1200
1195 O=S[2,12]
1200 PRINT "NUMERATOR COEFFICIENTS"
1205 FOR I=1 TO S[1,12]+1
1210 PRINT "N("I-1")="S[1,I]
1215 NEXT I
1220 PRINT "DENOMINATOR COEFFICIENTS"
1225 FOR I=1 TO S[2,12]+1
1230 PRINT "D("I-1")="S[2,I]
1235 NEXT I
1240 DISP "FREQ IN RAD/SEC(10↑?)--MIN,MAX";
1245 INPUT F0,F1
```

```
1250 I0=1
1255 F9=ABS(F1-F0)/30
1260 F5=F1+0.5*F9
1265 FOR F=F0 TO F5 STEP F9
1270 W=10↑F
1275 GOSUB 2000
1280 Q[1,I0]=LGT(W)
1285 Q[2,I0]=20*LGT(A9)
1290 Q[3,I0]=P9*57.29
1295 I0=I0+1
1300 NEXT F
1305 M0=0
1310 FOR I=1 TO 31
1315 IF ABS(Q[2,I]) <= M0 THEN 1325
1320 M0=Q[2,I]
1325 NEXT I
1330 DISP "LIST--Y OR N";
1335 INPUT P$
1340 IF P$="N" THEN 1385
1345 PRINT
1350 PRINT "BODE DATA--FREQ(RAD/SEC),FREQ(HZ),LOG AMP(DB),PHASE(DEG)"
1355 PRINT
1360 FLOAT 3
1365 FOR I=1 TO 31
1370 PRINT 10↑(Q[1,I]),(10↑(Q[1,I]))/(2*PI),Q[2,I],Q[3,I]
1375 NEXT I
1380 STANDARD
1385 FOR I=1 TO 31
1390 IF ABS(Q[3,I]) <= 180 THEN 1415
1395 IF Q[3,I]>180 THEN 1410
1400 Q[3,I]=Q[3,I]+360
1405 GOTO 1415
1410 Q[3,I]=Q[3,I]-360
1415 NEXT I
1420 DISP "PLOT--Y OR N";
1425 INPUT P$
1430 IF P$="N" THEN 1635
1435 I=2
```

```
1440 FOR J=1 TO 3
1445 IF ABS(M0) <= 1.75*I THEN 1475
1450 I=I*2
1455 IF I=80 THEN 1475
1460 NEXT J
1465 I=20
1470 GOTO 1440
1475 M0=I/2
1480 GOSUB 1665
1485 LABEL (*,1,1,0,7/10)
1490 C=2
1495 FOR I=1 TO 31
1500 X=(Q[1,I]-F0)*F2
1505 PLOT X,Q[2,I]*20/M0
1510 C=C+1
1515 IF C#3 THEN 1540
1520 C=0
1525 CPLOT -0.3,-0.3
1530 LABEL (*)"X"
1535 IPLOT 0,0
1540 NEXT I
1545 PEN
1550 C=2
1555 P0=Q[3,1]
1560 FOR I=1 TO 31
1565 X=(Q[1,I]-F0)*F2
1570 IF ABS(Q[3,I])>100 AND SGN(P0)#SGN(Q[3,I]) THEN 1580
1575 GOTO 1585
1580 PEN
1585 P0=Q[3,I]
1590 PLOT X,P0/3
1595 C=C+1
1600 IF C#3 THEN 1625
1605 C=0
1610 CPLOT -0.3,-0.3
1615 LABEL (*)"+"
1620 IPLOT 0,0
1625 NEXT I
1630 PEN
1635 IF D$="C" THEN 1655
1640 PRINT
1645 GOSUB 2255
1650 PRINT
1655 RETURN
1660 END
1665 REM******************************
1670 REM****BODE AXIS SUBROUTINE****
1675 REM******************************
1680 SCALE -35,165,-70,70
1685 LABEL (*,1.25,1.7,0,7/10)
```

```
1695 XAXIS 0,-1,0,150
1700 F7=0.1
1705 FOR J=F0 TO F1-0.1
1710 F7=F7*10
1715 FOR I=2 TO 10 STEP 2
1720 X=LGT(F7*I)
1725 PLOT X+F2,0,1
1730 IF I=10 THEN 1750
1735 IPLOT 0,1,-2
1740 IPLOT 0,-2,-1
1745 GOTO 1760
1750 IPLOT 0,2,2
1755 IPLOT 0,-4,-1
1760 NEXT I
1765 NEXT J
1770 LABEL (*,1.5,1.7,0,7/10)
1775 FOR F=F0 TO F1
1780 X=(F-F0)*F2
1785 IF X=0 THEN 1810
1790 PLOT X,0,1
1795 CPLOT -3.3,-1.5
1800 LABEL (1805)F
1805 FORMAT "10↑(",F2.0,")"
1810 NEXT F
1815 PLOT 75,-65,1
1820 LABEL (*,2,1.7,0,7/10)
1825 CPLOT -9.1,-0.6
1830 LABEL (*)"FREQUENCY (RAD/SEC)"
1835 PLOT 65,70,1
1840 B=LEN(T$)
1845 A=-((B+5)/2)+0.2
1850 CPLOT A,-0.6
1855 LABEL (*)"CASE "T$
1860 LABEL (*,1.5,1.7,0,7/10)
1865 PLOT 120,65,1
1870 LABEL (*)"X=AMPLITUDE"
1875 LABEL (*)"+=PHASE"
1880 YAXIS 0,20,-60,60
1885 YAXIS -20,15,-60,60
1890 FOR Y=-3 TO 3
1895 PLOT 0,Y*20,1
1900 CPLOT -4.5,-0.3
1905 LABEL (1910)Y*M0
1910 FORMAT 1F4.0
1915 NEXT Y
1920 FOR Y=-180 TO 180 STEP 90
1925 PLOT -20,Y/3,1
1930 CPLOT -5.5,-0.3
1935 LABEL (1940)Y
```

```
1940 FORMAT 1F3.0
1945 NEXT Y
1950 LABEL (*,2,1.7,PI/2,7/10)
1955 PLOT -10,0,1
1960 CPLOT -8.8,0
1965 LABEL (*)"LOG AMPLITUDE (DB)"
1970 PLOT -30,0,1
1975 CPLOT -5.3,0
1980 LABEL (*)"PHASE (DEG)"
1985 LABEL (*,1,1.0,7/10)
1990 RETURN
1995 END
2000 REM**************************************
2005 REM****AMPLITUDE & PHASE SUBROUTINE****
2010 REM**************************************
2015 R5=0
2020 R6=0
2025 I5=0
2030 I6=0
2035 FOR I=1 TO O+1
2040 P5=INT((I-1)/2)
2045 P6=(-1)↑P5
2050 IF (I-1-P5*2)=0 THEN 2070
2055 I5=I5+S[1,I]*P6*W↑(I-1)
2060 I6=I6+S[2,I]*P6*W↑(I-1)
2065 GOTO 2080
2070 R5=R5+S[1,I]*P6*W↑(I-1)
2075 R6=R6+S[2,I]*P6*W↑(I-1)
2080 NEXT I
2085 A9=SQR(I5↑2+R5↑2)/SQR(I6↑2+R6↑2)
2090 IF I5=0 AND R5=0 OR I6=0 AND R6=0 THEN 2140
2095 I1=I5
2100 R1=R5
2105 GOSUB 2155
2110 P7=P1
2115 I1=I6
2120 R1=R6
2125 GOSUB 2155
2130 P9=P7-P1
2135 RETURN
2140 W=W+0.01
2145 GOTO 2015
2150 END
2155 REM*************************
2160 REM****ARCTAN SUBROUTINE****
2165 REM*************************
2170 IF I1#0 AND R1#0 THEN 2230
2175 IF I1=0 AND R1>0 THEN 2220
2180 IF I1=0 AND R1<0 THEN 2210
2185 IF R1=0 AND I1>0 THEN 2200
```

```
2190 P1=1.5*PI
2195 RETURN
2200 P1=0.5*PI
2205 RETURN
2210 P1=PI
2215 RETURN
2220 P1=0
2225 RETURN
2230 P1=ATN(I1/R1)
2235 IF SGN(I1)=SGN(P1) THEN 2245
2240 P1=P1+PI
2245 RETURN
2250 END
2255 REM************************************
2260 REM****PHASE/GAIN MARGIN SUBROUTINE****
2265 REM************************************
2270 FIXED 2
2275 FOR I=1 TO 30
2280 IF SGN(Q[2,I])=SGN(Q[2,I+1]) THEN 2295
2285 P0=Q[3,I]-((Q[3,I+1]-Q[3,I])/(Q[2,I+1]-Q[2,I]))*Q[2,I]
2290 PRINT "AT 0 DB, PHASE="P0"DEGREES"
2295 NEXT I
2300 FOR I=1 TO 30
2305 IF SGN(Q[3,I])=SGN(Q[3,I+1]) THEN 2335
2310 IF ABS(Q[3,I])<125 THEN 2335
2315 Q1=180-ABS(Q[3,I])
2320 Q2=180-ABS(Q[3,I+1])
2325 A0=Q[2,I]+((Q[2,I+1]-Q[2,I])/(Q1+Q2))*Q1
2330 PRINT "AT 180 DEGREES, LOG AMPLITUDE="A0"DB"
2335 NEXT I
2340 STANDARD
2345 RETURN
2350 END
```

APPENDIX F

TR SUBROUTINE LISTING

```
1000 REM*****************************************
1005 REM****TIME RESPONSE EXECUTIVE SUBROUTINE*****
1010 REM*****************************************
1015 D$="N"
1020 DISP "ENTER CASE IDENTIFIER";
1025 INPUT T$
1030 PRINT "***************"
1035 PRINT "CASE "T$
1040 PRINT "***************"
1045 IF P[3,12]<P[2,12] THEN 1060
1050 D0=P[3,12]
1055 GOTO 1065
1060 D0=P[2,12]
1065 N0=P[1,12]
1070 STANDARD
1075 MAT S=ZER
1080 MAT T=ZER
1085 DISP "ENTER GAIN";
1090 INPUT Z
1095 PRINT
1100 PRINT "GAIN="Z
1105 PRINT
1110 FOR I=1 TO 11
1115 S[1,I]=Z*P[1,I]
1120 S[2,I]=Z*P[2,I]+P[3,I]
1125 NEXT I
1130 D=S[2,D0+1]
1135 FOR I=1 TO 11
1140 S[1,I]=S[1,I]/D
1145 S[2,I]=S[2,I]/D
1150 S[3,I]=S[2,I]
1155 NEXT I
1160 PRINT "NUMERATOR COEFFICIENTS"
1165 FOR I=1 TO N0+1
1170 PRINT "N("I-1")="S[1,I]
1175 NEXT I
1180 PRINT "DENOMINATOR COEFFICIENTS"
1185 FOR I=1 TO D0+1
1190 PRINT "D("I-1")="S[2,I]
1195 NEXT I
1200 MAT T=ZER
1205 C9=1
1210 O=D0
1215 Z9=1
1220 GOSUB 1965
1225 C9=C9-1
1230 N1=0
1235 FOR I=1 TO C9
1240 N1=N1+T[1,I]*T[2,I]
1245 NEXT I
```

```
1250 REDIM Q[N1,N1],A[N1],C[N1]
1255 MAT Q=ZER
1260 MAT C=ZER
1265 MAT A=ZER
1270 C1=1
1275 FOR J1=1 TO C9
1280 M=T[2,J1]
1285 IF T[1,J1]=2 AND M=1 THEN 1405
1290 IF T[1,J1]=2 AND M>1 OR T[1,J1]=1 AND M>3 THEN 1480
1295 Q[1,C1]=1
1300 FOR I=2 TO N1
1305 Q[I,C1]=T[3,J1]*Q[I-1,C1]
1310 NEXT I
1315 IF M=1 THEN 1395
1320 Q[1,C1+1]=0
1325 B=1
1330 FOR I=2 TO N1
1335 Q[I,C1+1]=T[3,J1]*Q[I-1,C1+1]+B
1340 B=T[3,J1]*B
1345 NEXT I
1350 IF M=2 THEN 1395
1355 Q[1,C1+2]=0
1360 B=0
1365 C=1
1370 FOR I=2 TO N1
1375 Q[I,C1+2]=T[3,J1]*Q[I-1,C1+2]+B
1380 B=T[3,J1]*B+2*C
1385 C=T[3,J1]*C
1390 NEXT I
1395 C1=C1+M
1400 GOTO 1470
1405 Q[1,C1]=1
1410 B1=0
1415 FOR I=2 TO N1
1420 Q[I,C1]=Q[I-1,C1]*T[3,J1]+B1*T[4,J1]
1425 B1=B1*T[3,J1]-Q[I-1,C1]*T[4,J1]
1430 NEXT I
1435 Q[1,C1+1]=0
```

```
1440 B1=1
1445 FOR I=2 TO N1
1450 Q[I,C1+1]=Q[I-1,C1+1]*T[3,J1]+B1*T[4,J1]
1455 B1=B1*T[3,J1]-Q[I-1,C1+1]*T[4,J1]
1460 NEXT I
1465 C1=C1+2
1470 NEXT J1
1475 GOTO 1490
1480 PRINT "YOU'RE KIDDING,YOU HAVE MULTIPLE COMPLEX OR REAL ROOTS!";
1485 END
1490 MAT Q=INV(Q)
1495 D=D0-N0
1500 FOR J1=0 TO N0-1
1505 A[D+J1+1]=S[1,N0-J1+1]
1510 IF J1=0 THEN 1530
1515 FOR J=0 TO J1-1
1520 A[D+J1+1]=A[D+J1+1]-S[2,D0-J1+J+1]*A[D+J+1]
1525 NEXT J
1530 NEXT J1
1535 A[Z9]=A[Z9]-S[1,1]/S[2,Z9]
1540 MAT C=Q*A
1545 C1=1
1550 FOR J=1 TO C9
1555 C8=T[1,J]*T[2,J]
1560 FOR I=1 TO C8
1565 T[4+I,J]=C[C1]
1570 C1=C1+1
1575 NEXT I
1580 NEXT J
1585 FLOAT 3
1590 PRINT
1595 PRINT "*****INVERSE LAPLACE TRANSFORM INFORMATION*****"
1600 PRINT "   FORM           R             I             A            B            C"
1605 FOR I=1 TO C9
1610 PRINT T[1,I];T[3,I];T[4,I];T[5,I];T[6,I];T[7,I]
1615 NEXT I
1620 PRINT
1625 STANDARD
```

AD-A051 468 AIR FORCE ARMAMENT LAB EGLIN AFB FLA F/G 9/2
STABILITY ANALYSIS PROGRAM FOR THE HEWLETT-PACKARD 9830A CALCUL--ETC(U)
JUL 75 K A GALE, R D GREEN, J M GONZALEZ
UNCLASSIFIED AFATL-TR-75-94 NL

2 OF 2
AD
A051 468

END
DATE
FILMED
4-78
DDC

```
1630 Q4=1
1635 FOR I=2 TO (Z9-1)
1640 Q4=Q4*I
1645 NEXT I
1650 REDIM Q[2,61]
1655 DISP "MAX. TIME ON PLOT=";
1660 INPUT T1
1665 T2=T1/60
1670 J=0
1675 Y1=0
1680 FOR T=0 TO T1 STEP T2
1685 IF Z9#1 THEN 1700
1690 Y=S[1,1]/(S[2,Z9]*Q4)
1695 GOTO 1705
1700 Y=(S[1,1]*T↑(Z9-1))/(S[2,Z9]*Q4)
1705 FOR I=1 TO C9
1710 IF ABS(T[3,I]*T) <= 226 THEN 1725
1715 D1=EXP(SGN(T[3,I]*T)*226)
1720 GOTO 1730
1725 D1=EXP(T[3,I]*T)
1730 IF T[1,I]#2 THEN 1745
1735 Y=Y+(T[5,I]*COS(T[4,I]*T)+T[6,I]*SIN(T[4,I]*T))*D1
1740 GOTO 1750
1745 Y=Y+(T[5,I]+T[6,I]*T+T[7,I]*T*T)*D1
1750 NEXT I
1755 J=J+1
1760 Q[1,J]=T
1765 Q[2,J]=Y
1770 IF ABS(Q[2,J])<Y1 THEN 1780
1775 Y1=ABS(Q[2,J])
1780 NEXT T
1785 PRINT
1790 PRINT "MAX TIME ON PLOT="T1
1795 PRINT "ABSOLUTE VALUE OF Y MAX="Y1
1800 IF S[2,1]=0 THEN 1825
1805 Y0=S[1,1]/S[2,1]
1810 F7=1
1815 PRINT "STEADY STATE VALUE OF Y="Y0
1820 GOTO 1840
1825 PRINT "NO STEADY STATE VALUE--"Z9-1"FREE INTEGRATORS"
1830 Y0=1
1835 F7=2
1840 PRINT
1845 IF Y0#0 THEN 1855
1850 Y0=1
1855 Y0=ABS(Y0)
1860 FOR M=0 TO 3
1865 J9=0.5*10↑M
1870 FOR I=1 TO 4
1875 J9=J9*2
```

```
1880 IF Y1 <= 1.5*J9*Y0 THEN 1905
1885 NEXT I
1890 NEXT M
1895 Y0=2*Y1/3
1900 GOTO 1910
1905 Y0=J9*Y0
1910 GOSUB 2465
1915 FOR I=1 TO 61
1920 PLOT Q[1,I]*60/T1,(Q[2,I]*20)/Y0
1925 NEXT I
1930 PEN
1935 DISP "NEW GAIN(G);TIME SCALE(T);NO(N)";
1940 INPUT D$
1945 IF D$="G" THEN 1075
1950 IF D$="T" THEN 1655
1955 RETURN
1960 END
1965 REM**********************************
1970 REM****POLYNOMIAL ROOT SUBROUTINE****
1975 REM**********************************
1980 R1=0
1985 R=0
1990 S=0
1995 IF O<2 THEN 2435
2000 IF S[3,1]#0 THEN 2050
2005 X=0
2010 Y=0
2015 T=1
2020 GOSUB 2670
2025 O=O-1
2030 FOR I=1 TO O+2
2035 S[3,I]=S[3,I+1]
2040 NEXT I
2045 GOTO 1995
2050 IF S#0 THEN 2065
2055 R=0
2060 GOTO 2075
2065 R=R/S
2070 S=1/S
2075 IF R1=0 THEN 2085
2080 R1=1/R1
2085 S1=5*10↑(-10)
2090 S[4,O+1]=S[3,O+1]
2095 S[5,O+1]=S[3,O+1]
2100 S[4,O+2]=0
2105 S[5,O+2]=0
2110 FOR I2=1 TO 20
2115 FOR I1=O TO 1 STEP -1
2120 S[4,I1]=S[3,I1]+R1*S[4,I1+1]
2125 S[5,I1]=S[4,I1]+R1*S[5,I1+1]
```

```
2130 NEXT I1
2135 IF (ABS(S[4,1]/S[3,1])-S1)>0 THEN 2185
2140 Y=0
2145 X=R1
2150 T=1
2155 GOSUB 2670
2160 O=O-1
2165 FOR I=1 TO O+1
2170 S[3,I]=S[4,I+1]
2175 NEXT I
2180 GOTO 2050
2185 IF S[5,2]#0 THEN 2200
2190 R1=R1+1
2195 GOTO 2205
2200 R1=R1-S[4,1]/S[5,2]
2205 FOR I=O TO 1 STEP -1
2210 S[4,I]=S[3,I]-R*S[4,I+1]-S*S[4,I+2]
2215 S[5,I]=S[4,I]-R*S[5,I+1]-S*S[5,I+2]
2220 NEXT I
2225 IF S[3,2]#0 THEN 2240
2230 IF (ABS(S[4,2]/S[3,1])-S1)>0 THEN 2250
2235 GOTO 2245
2240 IF (ABS(S[4,2]/S[3,2])-S1)>0 THEN 2250
2245 IF (ABS(S[4,1]/S[3,1])-S1) <= 0 THEN 2315
2250 T=S[5,2]-S[4,2]
2255 U=S[5,3]↑2-T*S[5,4]
2260 IF U#0 THEN 2280
2265 R=R-2
2270 S=S*(S+1)
2275 GOTO 2290
2280 R=R+(S[4,2]*S[5,3]-S[4,1]*S[5,4])/U
2285 S=S+(-S[4,2]*T+S[4,1]*S[5,3])/U
2290 NEXT I2
2295 S1=S1*10
2300 IF S1#5*10↑(-6) THEN 2110
2305 PRINT "TOLERANCE=5*10(-6),NO ROOTS"
2310 STOP
2315 G=R↑2-4*S
2320 IF G<0 THEN 2365
2325 X=-R/2+SQR(G)/2
2330 Y=0
2335 T=1
2340 GOSUB 2670
2345 X=-R/2-SQR(G)/2
2350 T=1
2355 GOSUB 2670
2360 GOTO 2385
2365 X=-R/2
2370 Y=SQR(-G)/2
2375 T=2
```

```
2380 GOSUB 2670
2385 O=O-2
2390 IF O=0 THEN 2455
2395 FOR I=O+1 TO 1 STEP -1
2400 S[3,I]=S[4,I+2]
2405 NEXT I
2410 IF O>2 THEN 2050
2415 IF O<2 THEN 2435
2420 R=S[3,O]/S[3,O+1]
2425 S=S[3,O-1]/S[3,O+1]
2430 GOTO 2315
2435 X=-S[3,O]/S[3,O+1]
2440 Y=0
2445 T=1
2450 GOSUB 2670
2455 RETURN
2460 END
2465 REM****************************************
2470 REM*****TIME RESPONSE AXES SUBROUTINE*****
2475 REM****************************************
2480 SCALE -20,80,-35,35
2485 XAXIS 0,10,0,60
2490 YAXIS 0,5,-30,30
2495 LABEL (*,1.5,1.7,0,7/10)
2500 FOR I=2 TO 6 STEP 2
2505 PLOT 10*I,0,1
2510 CPLOT -3.6,-1.3
2515 LABEL (2525)(I/6)*T1
2520 NEXT I
2525 FORMAT 1F7.2
2530 FOR I=-1 TO 1
2535 PLOT 0,I*20,1
2540 CPLOT -9.8,-0.3
2545 IF F7=2 THEN 2565
2550 LABEL (2555)I*J9*100
2555 FORMAT 1F9.0
2560 GOTO 2575
2565 LABEL (2570)I*Y0
2570 FORMAT 1E9.2
2575 NEXT I
2580 LABEL (*,2,1.7,0,7/10)
2585 PLOT 30,-30,1
2590 CPLOT -4.8,-0.6
2595 LABEL (*)"TIME (SEC)"
2600 PLOT 30,35,1
2605 B=LEN(T$)
2610 A=-((B+5)/2)+0.2
2615 CPLOT A,-0.6
2620 LABEL (*)"CASE "T$
2625 PLOT -10,0,1
```

```
2630 LABEL (*,2,1.7,PI/2,7/10)
2635 CPLOT -5.8,0
2640 IF F7=2 THEN 2655
2645 LABEL (*)"RESPONSE(%)"
2650 GOTO 2660
2655 LABEL (*)"RESPONSE (UNITS)"
2660 RETURN
2665 END
2670 REM********************************
2675 REM*****ROOT STORAGE SUBROUTINE*****
2680 REM********************************
2685 IF C9=1 THEN 2715
2690 FOR C8=1 TO C9-1
2695 IF T#T[1,C8] OR X#T[3,C8] OR Y#T[4,C8] THEN 2710
2700 T[2,C8]=T[2,C8]+1
2705 GOTO 2740
2710 NEXT C8
2715 T[1,C9]=T
2720 T[2,C9]=1
2725 T[3,C9]=X
2730 T[4,C9]=Y
2735 C9=C9+1
2740 IF T=1 AND X=0 THEN 2750
2745 RETURN
2750 Z9=Z9+1
2755 RETURN
2760 END
```

APPENDIX G

INVERSE LAPLACE TRANSFORM THEORY

This appendix explains the theory that was used to generate the inverse Laplace transform in the Time Response Subroutine.

The given system is described by the transfer function,

$$\frac{Y(s)}{R(s)} = \frac{b_m s^m + \ldots + b_1 s + b_o}{a_n s^n + \ldots + a_1 s + a_o} \quad (n \geq m) \tag{G-1}$$

or

$$Y(s)\,[a_n s^n + \ldots + a_1 s + a_o] = R(s)\,[b_m s^m + \ldots + b_1 s + b_o] \tag{G-2}$$

Taking the inverse Laplace transform of both sides yields,

$$a_n y^{(n)}(t) + \ldots + a_1 y^{(1)}(t) + a_o y(t) = b_m r^{(m)}(t) + \ldots + b_1 r^{(1)}(t) + b_o r(t) \tag{G-3}$$

where

$$y^{(n)} \triangleq \frac{d^n y(t)}{dt^n}$$

Although Equation (G-3) is valid (by definition) for all inputs and all initial conditions, certain problems arise when we attempt to take the derivative of a step input at $t = 0$. Thus, if $r(t) \triangleq u(t)$ and the b_i's $(i \geq 1)$ are not all zero, then Equation (G-3) needs additional interpretation and clarification. This particular case will be discussed in detail later, along with a derivation of the solution.

The general solution to any order linear differential equation with constant coefficients is given by

$$y(t) = y_h(t) + y_p(t) \tag{G-4}$$

where

$y_h(t)$ is the solution to the homogeneous equation, i.e., $a_n y^{(n)}(t) + \ldots + a_o y(t) = 0$

and

$y_p(t)$ is a unique, particular solution which is linearly independent of $y_h(t)$ and is non-zero if the right hand side of equation (G-3) is non-zero.

Consider first the general solution to the homogeneous equation. The general solution for single or multiply repeated real roots can be expressed as

$$Y_{hr}(t) = \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij}\, t^{j-1}\, e^{\lambda_i t} \tag{G-5}$$

where

λ_i is the distinct root of the characteristic equation,

I is the total number of different roots,

J is the multiplicity of the i^{th} root,

C_{ij} are the undetermined coefficients.

The general solution for complex roots can be expressed as

$$y_{hc}(t) = \sum_{i=1}^{I} e^{\alpha_i t}[A_i \cos\ \omega_i t + B_i \sin \omega_i t] \tag{G-6}$$

where

α_i is the real part of the i^{th} complex root pair,

ω_i is the imaginary part of the i^{th} complex root pair,

I is the total number of complex root paris,

A_i, B_i are the undetermined coefficients.

Therefore, the general homogeneous solution is the sum of Equations (G-5) and (G-6). The case of multiply repeated complex root pairs is so rare and cumbersome that it is omitted from the computer program.

Next consider the particular solution for the differential equation. The driving function for the Time Response Subroutine is a unit step applied at t = 0. The particular solution must satisfy the equation

$$a_n y_p^{(n)}(t) + \ldots + a_o y_p(t) = b_o u(t) + b_1 \delta(t) + \ldots + b_m \delta^{(m-1)}(t) \tag{G-7}$$

where

u(t) is the unit step function and $\delta^{(q)}(t)$ is the q^{th} derivative of the delta function,

which is the $(q + 1)^{th}$ derivative of the unit step function.

The problem with the unit step is that it is undefined at $t = 0$. Evaluating Equation (G-7) at $t = 0^+$ will alleviate the problem of an undefined unit step, and also, the delta function and its derivatives will be zero. However, evaluating the equation at $t = 0^+$ means that we will have to calculate new initial conditions for the system since the system was subjected to an infinite excitation at $t = 0$ and thus the system at $t = 0^+$ will not be at rest.

Therefore, Equation (G-7) becomes

$$a_n y_p^{(n)}(0^+) + \ldots + a_o y_p(0^+) = b_o \qquad \text{(G-8)}$$

where

$$u(0^+) = 1$$

$$\delta^{(q)}(0^+) = 0 \text{ for } q \geq 0$$

Assume a particular solution

$$y_p(t) = Kt^q \qquad \text{(G-9)}$$

where

q = the subscript of the coefficient of the lowest order, non-zero derivative in the left-hand side of Equation (G-8).

This general form of the particular solution is used because it is also valid for systems which have one or more poles at the origin.

Differentiation of Equation (G-9) yields

$$\begin{aligned}
&y_p(t) = Kt^q \\
&y_p^{(1)}(t) = qKt^{q-1} \\
&\bullet \\
&\bullet \\
&\bullet \\
&y_p^{(q)}(t) = q!\,K \qquad \text{(G-10)} \\
&y_p^{(q+1)}(t) = 0 \\
&\bullet \\
&\bullet \\
&\bullet \\
&y_p^{(n)}(t) = 0
\end{aligned}$$

When the set of equations in (G-10) is substituted into Equation (G-8) and the resulting equation is evaluated at $t = 0^+$ we have

$$a_q q!k = b_o$$

or

$$K = \frac{b_o}{a_q q!} \tag{G-11}$$

Therefore,

$$y_p(t) = \frac{b_o}{a_q} \frac{t^q}{q!} \tag{G-12}$$

Substituting Equations (G-5), (G-6), and (G-12) into Equation (G-4), differentiating the result (n - 1) times, and evaluating at $t = 0^+$ yields a set of equations,

$$\begin{aligned} y(0^+) &= y_p(0^+) + y_{hr}(0^+) + y_{hc}(0^+) \\ y^{(n-1)}(0^+) &= y_p^{(n-1)}(0^+) + y_{hr}^{(n-1)}(0^+) + y_{hc}^{(n-1)}(0^+) \end{aligned} \tag{G-13}$$

But

$$y_p^{(k)}(0^+) = 0 \text{ for } k \neq q$$

and

$$y_p^{(q)}(0^+) = b_o/a_q \quad \text{for } k = q$$

Therefore,

$$\begin{aligned} &y_{hr}(0^+) + y_{hc}(0^+) = y(0^+) \\ &\bullet \\ &\bullet \\ &\bullet \\ &y_{hr}^{(q)}(0^+) + y_{hc}^{(q)}(0^+) = y^{(q)}(0^+) - \frac{b_o}{a_q} \\ &\bullet \\ &\bullet \\ &\bullet \\ &y_{hr}^{(n-1)}(0^+) + y_{hc}^{(n-1)}(0^+) = y^{(n-1)}(0^+) \end{aligned} \tag{G-14}$$

where, in general, the $y^{(k)}(0^+)$'s are the initial conditions of the system at $t = 0^+$.

The final step prior to solving the set of equations (Equation G-14) for the undetermined coefficients associated with the homogeneous solutions is to determine the initial conditions at $t = 0^+$. Since we know that the initial conditions at $t = 0^-$ are all zero, we should be able to compute the state of the system at $t = 0^+$. Rewriting Equation (G-3) with $r(t) = u(t)$, yields

$$a_n y^{(n)}(t) + \ldots + a_o y(t) = b_o u(t) + b_1 \delta(t) + \ldots + b_m \delta^{(m-1)}(t) \qquad \text{(G-15)}$$

Let's integrate this equation n-times from $t = 0^-$ to t. The first integration yields

$$a_n \int_{0^-}^{t} y^{(n)}(t)dt + \ldots + a_o \int_{0^-}^{t} y(t)dt = b_o \int_{0^-}^{t} u(t)dt + b_1 \int_{0^-}^{t} \delta(t)dt + \ldots$$

$$+ b_m \int_{0^-}^{t} \delta^{(m-1)}(t)dt \qquad \text{(G-16)}$$

Integrating Equation (G-15) n times will yield n equations for the n derivatives $y(t)$, $\dot{y}(t), \ldots, y^{n-1}(t)$. Evaluating these n equations a $t = 0^+$ will result in the n initial conditions $y(0^+), \dot{y}(0^+), \ldots y^{n-1}(0^+)$. For example, the equation which results from integrating Equation (G-15) k times, when evaluated a $t = 0^+$ yields

$$a_n y^{(n-k)}(0^+) + \ldots + a_o y^{(-k)}(0^+) = b_o u^{(-k)}(0^+) + b_1 \delta^{(-k)}(0^+)$$

$$+ \ldots + b_o \delta^{(m-1-k)}(0^+) \qquad \text{(G-17)}$$

where, in general,

$$f^{(q)}(0^+) \triangleq f(t) \Big|_{t=0^+} \qquad \text{for } q = 0$$

$$f^{(q)}(0^+) \triangleq \frac{d^q}{dt^q} f(t)\Big|_{t=0^+} \qquad \text{for } q \geq +1$$

$$f^{(q)}(0^+) \triangleq \underbrace{\int_{0^-}^{0^+}\int_{0^-}^{t}\int_{0^-}^{t}}_{q \text{ times}} f(t)dt = f^{(q)}(0^+)\text{-}f^{(q)}(0^-) \qquad \text{for } q \leq -1$$

and

$f^{(q)}(0^-) = 0$ due to the fact that the system was at rest at $t = 0^-$.

Several observations can be made about the above equation.

(a) Since $y(t) = 0$ at $t = 0^-$ and must be finite at $t = 0^+$, then all integrals of $y(t)$ evaluated from $t = 0^-$ to $t = 0^+$ must be zero, i.e.,

$$y^{(q)}(0+) = 0 \qquad \text{for } q < 0$$

(b) The delta function terms on the right-hand side are

$$\delta^{(q)}(0^+) = 1 \qquad \text{for } q = -1$$

$$\delta^{(q)}(0^+) = 0 \qquad \text{for } q \geq 0$$

$$\delta^{(q)}(0^+) = \frac{t^{-q-1}}{(-q-1)!}\Big|_{t=0^+} = 0 \qquad \text{for } q \leq -2$$

(c) The unit step function on the right-hand side is

$$u(0^+) = 1 \qquad \text{for } q = 0$$

$$= \frac{t^{-q}}{(-q)!}\Big|_{t=0^+} = 0 \qquad \text{for } q \leq -1$$

Since the derivatives of the step function are expressed in terms of the δ - function, the case for $q \geq 1$ is not considered since no derivatives of u(t) will appear in any of the (n + 1) equations.

(d) Since evaluating $y^{(q)}(0^+)$ $(q \geq 1)$ involves a knowledge of $y^{(q-1)}(0^+)$, $y^{(q-2)}(0^+)$ . . . . $y(0^+)$, the proper procedure for evaluating the n-initial conditions involves first evaluating $y(0^+)$ using Equation (G-17) for k = n. Then $\dot{y}(0^+)$ can be evaluated using (G-17) with k = n - 1 and the previously computed value of $y(0^+)$. This procedure is continued until $y^{n-1}(0^+)$ is computed from (G-17), where k = 1.

Considering these observations, evaluating Equation (G-17) for several values of k yields

$$
\begin{aligned}
&y(0^+) = 0 \\
&\bullet \\
&\bullet \\
&\bullet \\
&y^{(n-m-1)}(0^+) = 0 \\
&y^{(n-m)}(0^+) = \frac{b_m}{a_n} \\
&y^{(n-m+1)}(0^+) = \frac{1}{a_n}\left[b_{m-1} - a_{n-1}\, y^{n-m}(0^+) \right] \\
&y^{(n-1)}(0^+) = \frac{1}{a_n}\left[b_1 - a_{n-1}\, y^{(n-2)}(0^+) \ldots . - a_1 y(0^+) \right]
\end{aligned}
\tag{G-18}
$$

The set of equations (G-18) can be written in a general form as:

$$
y^{(n-m+q)}(0^+) = \frac{1}{a_n}\left[b_{m-q} - \sum_{k-1}^{q} a_{n-k}\, y^{(n-m+q-k)}(0^+) \right] \tag{G-19}
$$

for

$$0 \leq q \leq m - 1$$

As shown in the set of equations, the value of the initial conditions for any derivative of order lower than (n - m) is zero, i.e.,

$$y^{(n-m+q)}(0^+) = 0 \qquad \text{for } q < 0.$$

Once the initial conditions have been evaluated using the above recursive relationship, the *final step involves determining the remaining constants (coefficients) when the set of equations* (G-14) is evaluated at $t = 0^+$. The most convenient method to accomplish this is to use the following matrix representation for n-simultaneous linear algebriac equations with n unknown coefficients:

$$\begin{bmatrix} A_{11} & \cdots\cdots\cdots & A_{1n} \\ \bullet & & \bullet \\ \bullet & & \bullet \\ \bullet & & \bullet \\ \bullet & & \bullet \\ A_{n1} & & A_{nn} \end{bmatrix} \begin{bmatrix} C_1 \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ C_n \end{bmatrix} = \begin{bmatrix} y(0^+)-y_p(0^+) \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ y^{(n-1)}(0^+)-y_p{}^{(n-1)}(0^+) \end{bmatrix} \qquad \text{(G-20)}$$

or

$$[A]\ [C] = [b]$$

where the column matrix [C] contains the undetermined coefficients. The coefficient matrix can be solved as follows:

$$[C] = [A]^{-1}\ [b]$$

After all of the coefficients have been determined, Equations (G-4), (G-5), (G-6), and (G-12) give the general solution for y(t).

INITIAL DISTRIBUTION

DDC	2
AUL (AUL/LSE-70-239)	1
ASD/ENYS	1
Hq USAF/SAMI	1
TAWC/TRADOCLO	1
AFATL/DL	1
AFATL/DLOSL	2
ASD/ENYEHM	1
Ogden ALC/MMNOP	2
AFWL/LR	2
AFATL/DLMA	50
AFIS/INTA	1

(The reverse of this page is blank)